

A Fully-Nonlocal Quasicontinuum Method to Model the Nonlinear Response of Periodic Truss Lattices

Thesis by
Gregory P. Phlipot

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy



CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2019
Defended May 16, 2019

© 2019

Gregory P. Philipot
ORCID: 0000-0003-2721-8678

All rights reserved except where otherwise noted

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Dennis Kochmann. After attending his computational mechanics course my first year at Caltech and seeing the high-quality research in his group, I knew that I wanted to work with him for my PhD. I was lucky enough to receive his excellent mentorship and guidance over the next four years, which consisted of many fruitful and though-provoking discussions. Even after accepting his current position at ETH and moving to Zürich, Dennis always made time in his busy schedule to be available to talk and guide me in the right direction, which I deeply appreciate. I would also like to thank my committee members, Dr. Michael Ortiz, Dr. Chiara Daraio, and Dr. Domniki Asimaki, for their time and effort spent participating in my defense, as well as their well-informed feedback.

I would like to express my gratitude to all of the professors who challenged me with meaningful research or coursework during my time at Arizona State and at Caltech. Specifically, I would like to thank Dr. Marc Mignolet, who convinced me to work on research during my undergraduate degree, persuaded me to pursue a PhD, and got me interested in research and the field of computational mechanics.

I would also like to thank all of the members of the Kochmann Research Group — both past and present — for all of the excellent discussions during group meetings and around the office. Alex Zelhofer and Ishan Tembhekar were especially welcoming when I joined the group and helped me with both research and non-research related matters. I would also like to extend a special thank you to the group members and friends at ETH, especially Abbas Tutcuoglu and Raphaël Glaesener, for making me feel at home during my two-month visit to Zürich. Danke.

There is no doubt in my mind that I would not have been able to complete my degree if it wasn't for all of my colleagues and friends at Caltech. Whether it was helping me finish up homework at midnight while sitting in our shared first-year office, keeping me company everyday at lunch, or letting go on one of our quasi-yearly trips to Las Vegas, they supported me along the way and helped me maintain the right amount of work-life balance needed to complete my degree.

Lastly, I would like to thank my parents, Sue and Mike Phlipot, for instilling the importance of education, inspiring me to become an engineer, and supporting me throughout my schooling.

This work was supported by a NASA Space Technology Research Fellowship.

ABSTRACT

We present a framework for the efficient, yet accurate description of general periodic truss networks based on concepts of the quasicontinuum (QC) method. Previous research in coarse-grained truss models has focused either on simple bar trusses or on two-dimensional beam lattices undergoing small deformations. Here, we extend the truss QC methodology to nonlinear deformations, general periodic beam lattices, and three dimensions. We introduce geometric nonlinearity into the model by using a corotational beam description at the level of individual truss members. Coarse-graining is achieved by the introduction of representative unit cells and a polynomial interpolation analogous to traditional QC. General periodic lattices defined by the periodic assembly of a single unit cell are modeled by retaining all unique degrees of freedom of the unit cell (identified by a lattice decomposition into simple Bravais lattices) at each macroscopic point in the simulation, and interpolating each degree of freedom individually. We show that this interpolation scheme accurately captures the homogenized properties of periodic truss lattices for uniform deformations. In order to showcase the efficiency and accuracy of the method, we compare coarse-grained simulations to fully-resolved simulations for various test problems, including: brittle fracture toughness prediction, static and dynamic indentation with geometric and material nonlinearities, and uniaxial tension of a truss lattice plate with a cylindrical hole. We also discover the notion of stretch locking — a phenomenon where certain lattice topologies are over-constrained, resulting in artificially stiff behavior similar to volumetric locking in finite elements — and show that using higher-order interpolation instead of affine interpolation significantly reduces the error in the presence of stretch locking in 2D and 3D. Overall, the new technique shows convincing agreement with exact, discrete results for a wide variety of lattice architectures, and offers opportunities to reduce computational expenses in structural lattice simulations and thus to efficiently extract the effective mechanical performance of discrete networks.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] Gregory P. Phlipot and Dennis M. Kochmann. A quasicontinuum theory for the nonlinear mechanical response of general periodic truss lattices. *Journal of the Mechanics and Physics of Solids*, 124:758 – 780, 2019. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2018.11.014>. URL <http://www.sciencedirect.com/science/article/pii/S0022509618304071>. Gregory P. Phlipot participated in the formulation and implementation of the method, performed all simulations, and co-wrote the manuscript.

TABLE OF CONTENTS

Acknowledgements	iii
Abstract	v
Published Content and Contributions	vi
Bibliography	vi
Table of Contents	vii
List of Illustrations	ix
List of Tables	xvii
Chapter I: Introduction	1
1.1 Overview of Microstructured Materials	1
1.2 Outline of this Thesis	4
1.3 Finite-Sized Truss Structures	5
1.4 Brief Overview of Homogenization Theory	7
1.5 Properties of Periodic Truss Lattices	10
Chapter II: Numerical Modeling of Truss Structures	16
2.1 High-Fidelity RVE Characterization	16
2.2 Fully-Resolved Simulations	17
2.3 Numerical Homogenization	18
2.4 Quasicontinuum Method	20
2.5 Shortcoming of Existing Methods	23
Chapter III: Fundamentals of the Quasicontinuum Method	26
3.1 Initial Problem Statement	26
3.2 Modeling Individual Truss Members	29
3.3 History-Dependent Problems	36
3.4 The Quasicontinuum Method	43
3.5 Adaptive Refinement	51
3.6 Comparison with FE^2 and classical homogenization	52
Chapter IV: Applications	54
4.1 Elastic Constants	54
4.2 Boundary Layer Method	60
4.3 Indentation	66
4.4 Dynamic Indentation	70
4.5 Hole in Plate Made of an Elasto-Plastic Truss Lattice	75
Chapter V: Higher-Order Quasicontinuum Techniques and Stretch Locking	80
5.1 Stretch Locking	80
5.2 Higher-Order Interpolation and Summation Rules	82
5.3 Performance of Higher-Order Methods	89
Chapter VI: Numerical Considerations	101
6.1 Structure of a Simulation	101
6.2 Computational Costs	111

Chapter VII: Discussion and Outlook	117
7.1 Accomplishments	117
7.2 Future Work	119
Bibliography	123
Appendix A: Effective Elastic Constants of Periodic Truss Lattices . .	137
Appendix B: Calculating Sample Unit Cell Weights in 3D	139

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Scanning electron micrographs of natural materials with a mechanical microstructure, taken from [48]	1
1.2 Examples of periodic truss lattices: Ultralight hollow metallic microlattice [93] (left), solid carbon lattice reprinted by permission from Springer Nature: Nature Materials [12], copyright 2016 (center), and a hierarchical lattice reproduced from [70](right).	3
1.3 Examples of rigid and non-rigid truss structures. On the left, a rigid truss structure satisfying the necessary conditions for static and kinematic determinacy with no states of self-stress or mechanisms. In the center is a truss structure that does not satisfy Maxwell's inequality, and thus has a mechanism (shown in dotted lines). On the right, a truss structure that satisfies the necessary conditions for static and kinematic determinacy, but has both a mechanism and a state of self-stress.	7
1.4 Two-scale nonlinear elasticity boundary value problem with essential boundary conditions and applied tractions \mathbf{T} . As the characteristic size of the unit cell ε tends towards zero, homogenization attempts to find a smooth homogenized energy density $W(\nabla \mathbf{u})$ to approximate the highly oscillatory energy density of the	8
1.5 The unit cell problem with periodic boundary conditions. The average deformation gradient of the macroscale \mathbf{F} is applied to the unit cell, and the fine-scale correction \mathbf{v} is determined by minimizing the energy of the unit cell over all periodic corrections.	9
1.6 Categorizing lattices and bending- or stretching-dominated lattices and as Bravais lattices or multilattices.	12
1.7 Depiction of the boundary layer method for a cellular material. Reprinted by permission from Springer Nature: International Journal of Fracture [94], copyright 2001.	14

1.8	Elastic moduli and yield strength of various materials. Note that the predicted stiffness and strength of lattices made of aluminum or carbon fiber reinforced polymers (CRFP) populate otherwise empty regions of the material space. Reproduced from Fleck et al. [40] with permission of the copyright holder, The Royal Society.	15
2.1	Computed homogenized stiffness of solid (left) and hollow shell (right) octet truss lattices using continuum-level finite element analysis (with the commercial code Abaqus), and structural Euler-Bernoulli and Timoshenko beams for various relative densities. Reprinted from [71].	17
2.2	Simulation of uniaxial tension of a notched hollow octet truss lattice modeled using shell finite elements to investigate the insensitivity of truss lattices to flaws. Reprinted from [73].	18
2.3	Pictorial description of numerical homogenization reprinted from Matouš et al. [67] with permission of the copyright holder, Elsevier. \mathbf{F}_M represents the macroscopic deformation gradient, and \mathbf{S}_M and \mathbf{C}_M represent the macroscopic stress and tangent matrices that are returned to the macroscale after solving the RVE problem.	19
2.4	The traditional QC method for atomic lattices (left) and the extension to truss lattices (right).	21
2.5	The QC method paired with aspects of the extended finite element method to model the crack propagation (crack path shown in red) in a cross-braced truss lattice. Reprinted from Rokoš et al. [88] with permission from the copyright holder, Elsevier .	24
3.1	(a) Example definition of a unit cell Ω_u based on its node(s) (solid circle) and its connecting beams \mathcal{E}_u^n (in this case $\mathcal{E}_u = \emptyset$). Further shown are examples of (b) a simple Bravais lattice (diamond) and (c) a multilattice (hexagon, composed of two simple Bravais lattices) and their corresponding unit cells and unit cell dofs. The dofs of all nodes inside each unit cell are retained with φ denoting the generalized dofs of a truss node. .	27

3.2	Kinematics of the 3D corotational beam: the linear Euler Bernoulli beam rotates with the element frame \mathbf{R}^b , and has small angle deflections at each node. Superscripts r , u , d , and b stand for the (r)eference, (u)ndeformed, (d)eformed, and (b)eam configuration, respectively.	33
3.3	Example stress-strain curve for the plasticity model with isotropic hardening. Reprinted by permission from Springer Nature: Springer Computational Inelasticity by J.C. Simo and T.J.R. Hughes, copyright 2000 [97].	39
3.4	Illustration of the interpolated dofs and the recovery of nodal dofs within individual unit cells from the interpolation: (a) shows a mesh of coarse-grained/macrosopic elements whose vertices are repUCs that carry dofs φ . (b) Within an element K , the dofs φ are interpolated by evaluating the shape functions at each unit cell location, e.g. \mathbf{X}_i or \mathbf{X}_j (in the reference configuration). (c) Dofs $\{\varphi^0, \varphi^1\}$ of the two nodes within each unit cell are recovered from the interpolated repUC dofs.	44
3.5	First and second-order summation rules. The first-order summation rule (left) shows the vertex sampling unit cells (blue) and the local inner sampling unit cell (red). The second-order summation rule (right) also has edge sampling unit cells (green).	48
4.1	Unit cells of lattices with (a) square, (b) hexagon, (c) kagome, (d) triangle, (e) star-shaped 2D, (f) cubic, (g) cuboctahedron, and (h) BCC architecture.	55
4.2	Representative, randomly coarse-grained truss QC simulations in 2D and 3D for the elastic constant calculation: (left) a coarse-grained hexagon lattice with 30% repUC density and (right) a bcc lattice with 10% repUC density. Affine boundary conditions are applied to the translational dofs on the outer faces of the block, and the energy of all sample unit cells inside the yellow box is used to calculate the elastic modulus (the large region in between removes any boundary layer effects).	57

4.3	Errors in the linear elastic bulk, shear, and uniaxial moduli of 2D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with the first-order (<i>left</i>) and second-order (<i>right</i>) optimal summation rule and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.	58
4.4	Errors in the linear elastic bulk, shear, and uniaxial moduli of 3D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with the first-order (<i>left</i>) and second-order (<i>right</i>) optimal summation rule and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.	59
4.5	Coarse-grained kagome lattice in 2D with a pre-existing crack undergoing mode-I loading, simulated by truss QC with adaptive mesh refinement. The colors correspond to the (normalized) maximum tensile stresses in the beams.	62
4.6	Fully-resolved simulation of the 2D kagome lattice undergoing mode 1 loading. The colors correspond to the (normalized) maximum tensile stresses in the beams.	63
4.7	Convergence of the maximum strut tensile stress near the notch tip (<i>left</i>) and of the total energy (<i>right</i>) as compared to the fully-resolved simulation for coarse-grained truss QC fracture simulations of a (<i>top</i>) kagome lattice, (<i>bottom</i>) triangle lattice in 2D. Insets visualize the adaptively refined QC representation at the indicated repUC densities.	64
4.8	Convergence of the maximum strut tensile stress near the notch tip (<i>left</i>) and of the total energy (<i>right</i>) as compared to the fully-resolved simulation for coarse-grained truss QC fracture simulations of a (<i>top</i>) hexagon lattice, (<i>bottom</i>) star-shaped lattice in 2D. Insets visualize the adaptively refined QC representation at the indicated repUC densities.	65

4.9	Initial geometry for the fully-resolved (<i>left</i>) and coarse-grained (<i>right</i>) 2D kagome lattice before indentation.	67
4.10	(a) Normalized load-displacement curves for the indentation into a 2D kagome lattice, (b) the repUC density as the simulation progresses, and the final deformed configurations of the (c) fully-resolved Euler-Bernoulli beam, (d) fully-resolved corotational beam, and (e) adaptively refined corotational beam simulations. The colors of beams correspond to the maximum tensile stress in each beam normalized by Young's modulus.	69
4.11	(a) Initial geometry for the fully-resolved and (b) coarse-grained truss QC of a 3D kagome lattice before indentation.	70
4.12	Normalized load-displacement curve for the indentation of a 3D kagome lattice (<i>top left</i>), the repUC density as the simulation progresses (<i>top right</i>), and the slices of the final deformed configurations of the fully-resolved (<i>bottom left</i>) and adaptively refined (<i>bottom right</i>) simulations with average axial strain contours. .	71
4.13	Geometry of the coarse-grained dynamic indentation simulation. The left shows the initial coarse-grained truss lattice with the mesh and sample unit cells.	72
4.14	Snapshots of the truss lattice: coarse-grained (left) and fully-resolved (right). Colors correspond to the maximum stress in each beam, and the broken beams are not shown.	74
4.15	Magnitude of the acceleration in the coarse-grained (left) and fully-resolved (right) simulations at various time steps before failure of the truss lattice. The elastic waves can be seen artificially bounding off mesh boundaries in the coarse-grained simulations. .	76
4.16	Cylindrical hole in octet lattice plate. The fully-resolved truss lattice (left) and the coarse-grained mesh (right)	77
4.17	Average stress-strain relation for the octet truss hole-in-plate extension simulations with first- and second-order summation rules, compared with the fully-resolved simulation. The insets are colored by the axial stress in each bar in the specimen at macroscopic strain values of $\varepsilon = 0.08$ and $\varepsilon = 0.20$	78

4.18	Plastic strains near the cylindrical hole at 20% macroscopic strain for the coarse-grained simulations with first-order summation rule (<i>top</i>), second-order summation rule (<i>middle</i>), and the fully-resolved simulation (<i>bottom</i>). The maximum plastic strains are 0.597 (coarse-grained, first-order summation) 0.632 (coarse-grained, second-order summation), and 0.627 (fully-resolved).	79
5.1	Traditional Cook's membrane geometry (<i>left</i>) and the corresponding hexagon lattice membrane (<i>right</i>) with each truss member having a length of $L = 1$	81
5.2	Example meshes for the coarse-grained truss lattice Cook's membrane simulations. Full resolution (<i>top</i>), coarsening of $n = 4$ (<i>bottom left</i>), and coarsening of $n = 8$ (<i>bottom right</i>) of the 2D kagome lattice.	83
5.3	Normalized strain energy of truss lattice versions of Cook's membrane problem for various 2D truss topologies with relative density $\bar{\rho} = 1\%$. Insets show the deformed fully-resolved and coarse-grained (with repUC density of $1/9$) simulations of the hexagon(<i>top</i>) and star-shaped (<i>bottom</i>) lattices, colored by the displacements normalized by the maximum displacement of the fully-resolved truss.	84
5.4	Strain energy of the hexagon and star-shaped lattices at relative densities of $\bar{\rho} = 10\%$ and $\bar{\rho} = 1\%$	85
5.5	An example of higher-order interpolation QC used to model bending of planar lattices (<i>left</i> , reprinted from [16] with permission of the copyright holder, Elsevier), and an example of a quadratic tetrahedron in the QC method used to model a "lattice continuum" (<i>right</i>).	85
5.6	Example of the proposed higher-order interpolation strategy applied to a square lattice. The underlying truss lattice (grey) is approximated using the repUCs (blue) connected by the mesh (black).	88

5.7	Errors in the linear elastic bulk, shear, and uniaxial moduli of 2D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with quadratic interpolation paired with quadrature (<i>left</i>) and optimal (<i>right</i>) summation rules and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.	90
5.8	Errors in the linear elastic bulk, shear, and uniaxial moduli of 3D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with quadratic interpolation paired with quadrature (<i>left</i>) and optimal (<i>right</i>) summation rules and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.	91
5.9	Normalized strain energy of the Cook's membrane simulations of the hexagon (<i>top</i>) and star-shaped (<i>bottom</i>) lattices with relative density of $\bar{\rho} = 1\%$. The insets show the deformed membranes and are colored by the magnitude of displacements (normalized by the maximum displacement of the fully-resolved simulation).	93
5.10	Convergence of the strain energy and maximum tensile stress in a strut near the notch tip for the hexagon and star-shaped lattice coarse-grained boundary layer simulations using the quadratic interpolation and the optimal (blue) and exact (orange) summation rules.	96
5.11	A lattice block with distributed shear load placed on the top. The cube is made by tessellating either the tetrakaidecahedron (<i>top right</i>) or the 3D star-shaped (<i>bottom right</i>) unit cell.	97
5.12	Strain energy in the coarse-grained QC simulations of the tetrakaidecahedron lattice block undergoing a distributed shear load across the top surface. The insets show the deformed configurations of all of the sampling unit cells, and are color-coded based on the displacements.	98

5.13	Strain energy in the coarse-grained QC simulations of the 3D star-shaped lattice block undergoing a distributed shear load across the top surface. The insets show the deformed configurations of all of the sampling unit cells, and are color-coded based on the displacements.	99
6.1	Program flow chart.	102
6.2	Examples of sampling truss elements (with small nodes) and their corresponding energy sampling elements (with larger nodes). The colors of the truss members match the nodes of the mesh needed to compute the energy of the truss member (i.e. the repUCs with nonzero shape functions at the truss member nodes). The number of nonzero shape functions can vary - examples of truss members with 1 (green), 2 (yellow), 3 (red), and 4 (blue) repUC dependencies are shown.	105

LIST OF TABLES

<i>Number</i>	<i>Page</i>
6.1 Time (in seconds) spent assembling and solving the lattice block shear simulations in Section 5.3.4 with quadratic interpolation. Simulations were performed on a machine with an Intel Xeon CPU E5-2650 processor using 12 cores with 1 MPI process per core.	115
6.2 Time (in seconds) spent in different parts of the code for the dynamic indentation simulations in Section 4.4. "Total Solve" represents the total time in the nonlinear solver (i.e. assembling systems, solving linear systems, and performing line searches). Simulations were performed serially on a machine with an Intel Xeon CPU E5-2650 processor.	116

INTRODUCTION

1.1 Overview of Microstructured Materials

Many materials that appear to be homogeneous — both natural and engineered — have a *mechanical microstructure* that greatly influences their macroscopic properties. All materials have some sort of microstructure (e.g. polymer networks, atomic grain structures), but by mechanical microstructure, we mean structural features on a mesoscale larger than the typical microstructural scales, but on a much smaller scale than the macroscopic size of an object. Two quintessential examples of naturally occurring materials with mechanical microstructure are wood and bone (see Fig. 1.1), but the properties of other natural materials are also greatly influenced by their microstructure. The microstructure in wood comes in the form of fibers, which give it its characteristic anisotropy. On the other hand, bone has a seemingly random porous microstructure, which can lead to isotropic macroscopic properties.

Many *engineered* materials also make use of microstructure to obtain certain desirable macroscopic properties. The most common of these materials are engineered foams. Foams can be made out of a variety of constituent materials, but they are characterized by their cellular nature and random microstructure.

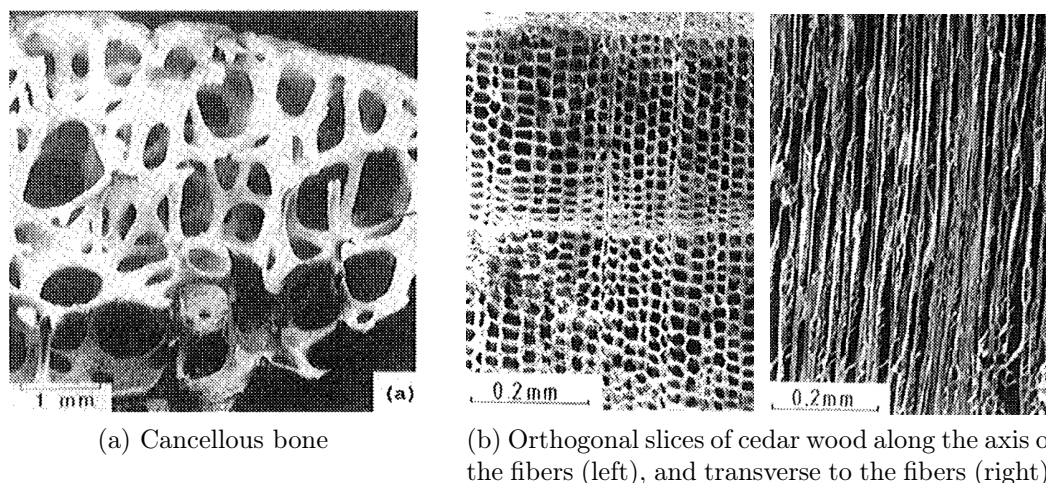


Figure 1.1: Scanning electron micrographs of natural materials with a mechanical microstructure, taken from [48]

Foams are used in many products to achieve a wide variety of reasons, including

- **Cars.** Car bumpers utilize plastic foams to maximize energy dissipation in collisions.[95].
- **Houses.** Plastic foams are the most common thermal insulator used in buildings[25].
- **Aerospace vehicles.** The strong and lightweight nature of foams make them ideal core materials for sandwich panels commonly used in airplanes [50].
- **Mattresses.** Memory foam (developed by NASA in the 1970s to improve seat cushioning) is commonly used to make comfortable beds [27].

An excellent overview of mechanical, acoustic, and electrical properties of various microstructured materials (including foams, honeycombs, and natural materials) was presented in Gibson and Ashby [48].

Even though the manufacturing of foams and other materials with random microstructure has improved and enabled engineering feats like landing a rover on Mars¹, manufacturing materials with a designed, ordered microstructure has been out of reach until recently. The recent advances in additive manufacturing techniques have enabled the creation of materials with an arranged microstructure — commonly referred to as metamaterials² — giving engineers the ability to design materials with virtually any microstructure.

One popular class of mechanical metamaterials are periodic truss lattices. In part due to their manufacturability, and in part due to some of their desirable theoretical properties, periodic truss lattices have garnered significant interest in the past decade. Notably, a metal hollow microlattice manufactured by HRL Laboratories in 2011 pushed the boundaries of ultralight materials by creating a metamaterial that is 99.99% air [93]. There has also been pioneering work at smaller scales. Additive manufacturing techniques have been used to

¹NASA’s Mars Science Laboratory spacecraft carried the Curiosity rover to the martian surface in 2012 using a Phenolic Impregnated Carbon Ablator (PICA) heat shield, which inherits its desirable thermal properties in part from the microstructure of the material.

²The name *metamaterials* was originally used to refer to microstructured materials with novel optical or electromagnetic properties, but has since been adopted by other fields to refer to microstructured materials with other desirable properties (e.g. mechanical, acoustic, or thermal).

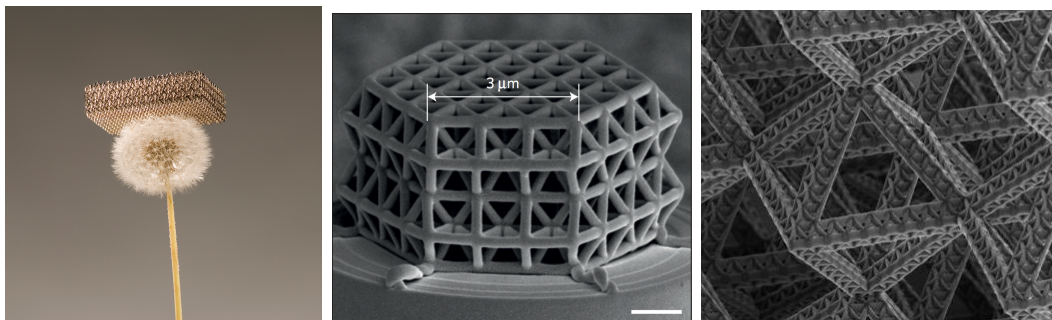


Figure 1.2: Examples of periodic truss lattices: Ultralight hollow metallic microlattice [93] (left), solid carbon lattice reprinted by permission from Springer Nature: Nature Materials [12], copyright 2016 (center), and a hierarchical lattice reproduced from [70](right).

manufacture polymer lattices with sub-micrometer resolution [66]. Furthermore, these polymer lattices have been coated with a thin layer of ceramic and the polymer has been etched out, leaving an ultralight ceramic hollow truss lattice with wall thicknesses as low as 5 nanometers [69]. In addition to the standard periodic truss lattices, there has been interest in hierarchical truss lattices [70, 115] (truss lattices made of truss lattices) as hierarchical microstructures are seen in nature.

The ability to design and manufacture these multiscale materials leads directly to two questions:

1. Given a certain microstructure and the properties of the constituent material, what are the effective properties at the macroscale?
2. Given desired macroscopic properties, what microstructure and constituent material can be used to achieve the properties?

The first question is hereby referred to as the *forward* problem since it requires computing quantities given known inputs. Conversely, the second is referred to as the *inverse* problem because it requires finding the inputs that give a desired output. Since solving the inverse problem requires solving the forward problem, the decades-old field of extracting the effective macroscopic response of microstructured materials has recently been revitalized by the advancements in manufacturing. The theoretical foundation for the micromechanical modeling of metamaterials probably began with Hill [52] calculating of the effective, homogenized moduli of periodic linear elastic composites. Various extensions

of homogenization theory followed, including probabilistics and wave propagation [21] as well as nonlinearity [46].

While there has been significant work towards answering the forward problem, the ability to predict the general nonlinear response of metamaterials (e.g. buckling of truss members, densification, failure of microstructure components, etc.) is still a challenge. This results in an expensive and slow design cycle relying on excessive manufacturing and testing to span the design space looking for microstructures that give desired properties. Ideally, this process could be expedited using theoretical or computational models, such that solving the forward problem becomes inexpensive and fast. Then these models could be used to span the design space of microstructures, and augment the design process to solve the inverse problem. This is the motivation for the research in this thesis.

1.2 Outline of this Thesis

The goal of this thesis is to develop a multiscale modeling technique that can be used to efficiently and accurately model the nonlinear mechanical response a particular type of metamaterial: periodic truss lattices. In the remainder of this chapter, we give an overview of truss lattices including some previous relevant theoretical results, and the experimental literature. Chapter 2 contains the fundamentals of numerical modeling of truss lattices and provides details on popular numerical methods and their shortcomings, motivating the research in this thesis: an extension of the quasicontinuum (QC) method to model complex trusses. Chapter 3 outlines the QC method and formulates the extension of the method which represents the bulk of the research. Chapter 4 is composed of multiple test cases to show the usefulness of the previously formulated QC method. One of the test cases shows a disagreement between the QC method and fully-resolved simulations when it comes to predicting the response of some bending-dominated lattices. This motivates Chapter 5, where the notion of *stretch locking* is investigated, and the previous QC formulation is modified by using higher-order interpolation functions in order to mitigate this phenomenon, leading to a more accurate model. In Chapter 6, the computational performance of the method is discussed, and the thesis concludes in Chapter 7.

1.3 Finite-Sized Truss Structures

Understanding the mechanical properties of trusses starts with understanding the differences between the bending stiffness and axial stiffness of a beam. In an undergraduate mechanics course, we learn that the tip displacements of a cantilevered beam corresponding to a tip load F in the axial and transverse directions are

$$\Delta x = \frac{FL}{EA} \quad \text{and} \quad \Delta y = \frac{FL^3}{3EI}, \quad (1.1)$$

respectively, where L is the length of the beam, E is the Young's modulus of the material, A is the cross-sectional area, and I is the moment of inertia. If the beam has a characteristic cross-section dimension r , the area of the beam in three dimensions is proportional to r^2 , while the moment of inertia is proportional to r^4 . This means that the stretching and bending stiffness per unit length of a beam in 3D satisfy

$$k_{stretch}^{(3D)} \sim \left(\frac{r}{L}\right)^2 \quad \text{and} \quad k_{bend}^{(3D)} \sim \left(\frac{r}{L}\right)^4. \quad (1.2)$$

Similar arguments for two-dimensional beams (where the out-of-plane thickness of the beam is constant and independent of r) lead to

$$k_{stretch}^{(2D)} \sim \left(\frac{r}{L}\right) \quad \text{and} \quad k_{bend}^{(2D)} \sim \left(\frac{r}{L}\right)^3. \quad (1.3)$$

Since trusses are assumed to be composed of slender beams (i.e. beams with $r/L \ll 1$), the axial, or stretching stiffness can be orders of magnitude larger than the bending stiffness.

For that reason, the concept of kinematically and statically determinate structures is important to understanding the mechanical properties of trusses. A truss structure is said to be *kinematically determinate* if there are no kinematically admissible deformations (excluding rigid body modes) such that no truss member changes its length. Conversely, *kinematically indeterminate* trusses are defined as structures where there exists at least one kinematically admissible deformation mode (excluding rigid body modes) such that no truss members change length. Similarly, a structure is said to be *statically determinate* if there exist no (nonzero) stress states such that the net force on all nodes is zero, and a *statically indeterminate* structure has at least one nonzero stress state that results in zero forces on all nodes.

If the truss is statically indeterminate, the states of stress that produce a net zero force on all nodes are called *states of self-stress*. If a truss is kinematically

indeterminate, the deformation modes where no truss members change length are called *mechanisms*. They are called mechanisms since the truss is able to deform purely through bending of beams, which is orders of magnitude less stiff than the stretching deformation modes. In fact, it is common to analyze truss structures as pin-jointed bars, where the mechanisms are treated as zero-energy modes. It is therefore common to refer to kinematically determinate structures as *rigid* trusses, while kinematically indeterminate structures are *non-rigid*.

A detailed formulation of statically and kinematically determinate truss structures is given in Pellegrino and Calladine [80], and is summarized here. Given a truss structure with n nodes and b beams, a matrix $\mathbf{A} \in \mathbb{R}^{dn \times b}$ in d dimensions, called the equilibrium matrix, can be used to relate the displacements of the nodes, \mathbf{u} , to the elongations of the beams, \mathbf{e} , according to

$$\mathbf{A}^T \mathbf{u} = \mathbf{e}. \quad (1.4)$$

By using the principle of virtual work, the matrix \mathbf{A} can also be used to relate the tensions in each bar, \mathbf{t} , and the forces at each node, \mathbf{f} , via

$$\mathbf{A} \mathbf{t} = \mathbf{f}. \quad (1.5)$$

By the definitions of states of self-stress and mechanisms given above, the states of self-stress satisfy the equation $\mathbf{A} \mathbf{t} = \mathbf{0}$, while the mechanisms satisfy $\mathbf{A}^T \mathbf{u} = \mathbf{0}$. In other words, the states of self-stress form the nullspace of \mathbf{A} , while the mechanisms form the left nullspace of \mathbf{A} .

Therefore, the rigidity of a truss can be determined by the dimensionality of the left nullspace of \mathbf{A} . If $\dim(\text{null}(\mathbf{A}^T)) = 3$ in 2D or $\dim(\text{null}(\mathbf{A}^T)) = 6$ in 3D, then the only mechanisms are the rigid body modes, and the truss structure is rigid. The dimension of the left nullspace of a matrix is at least as large as the difference between the number of rows and columns in the matrix. This leads directly to Maxwell's relation for the necessary condition for a truss lattice to be rigid [44],

$$b \geq 2d - 3 \quad \text{in 2D, and} \quad b \geq 3d - 6 \quad \text{in 3D.} \quad (1.6)$$

Similar analysis with the states of self-stress lead to the necessary condition for a truss to be statically determinate (i.e. there are no states of self-stress), or

$$b \leq 2d - 3 \quad \text{in 2D, and} \quad b \leq 3d - 6 \quad \text{in 3D.} \quad (1.7)$$

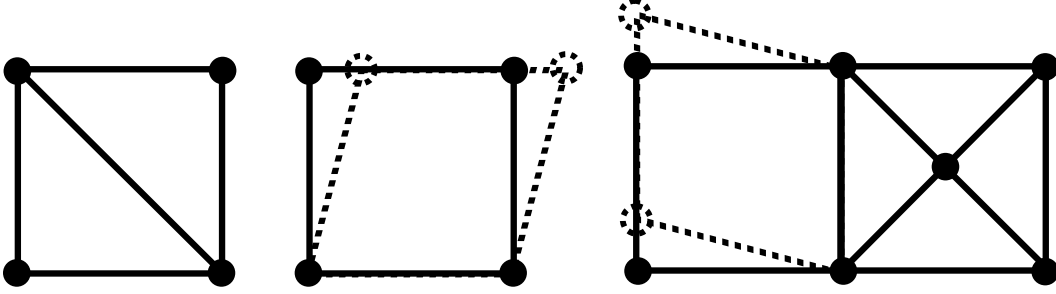


Figure 1.3: Examples of rigid and non-rigid truss structures. On the left, a rigid truss structure satisfying the necessary conditions for static and kinematic determinacy with no states of self-stress or mechanisms. In the center is a truss structure that does not satisfy Maxwell’s inequality, and thus has a mechanism (shown in dotted lines). On the right, a truss structure that satisfies the necessary conditions for static and kinematic determinacy, but has both a mechanism and a state of self-stress.

The combination of the two equations suggests that a necessary (but not sufficient) condition for a truss to be both statically and kinematically determinate is when the inequalities are satisfied as equalities. See Fig. 1.3 for examples of rigid and non-rigid truss structures. For more information on the equilibrium matrix, and how the four fundamental subspaces of \mathbf{A} relate to states of self-stress and mechanisms, see Pellegrino and Calladine [80].

1.4 Brief Overview of Homogenization Theory

Due to the multiscale nature of periodic truss lattices, homogenization techniques are used to understand the response of the effective material. One of the most important homogenization problems for understanding the quasistatic response of metamaterials is that of the two-scale nonlinear elasticity problem. Since the conclusions drawn from the two-scale homogenization problem are so important (and will be referenced throughout the thesis) we take the time here to outline the problem, go over key results, and show how the theory can be applied to periodic truss lattices.

The macroscale boundary value problem of a heterogeneous body Ω with statistically homogeneous microstructure — represented by a representative volume element (RVE) Ω_{UC} — aims to identify the deformation mapping or displacement field as the minimizer of the total potential energy

$$\mathcal{I}[\mathbf{u}^\epsilon] = \int_{\Omega} W\left(\frac{\mathbf{x}}{\epsilon}, \nabla \mathbf{u}^\epsilon\right) dV - \mathcal{L}(\mathbf{u}^\epsilon) \quad (1.8)$$

subject to essential boundary conditions on $\partial\Omega_1$ and traction boundary con-

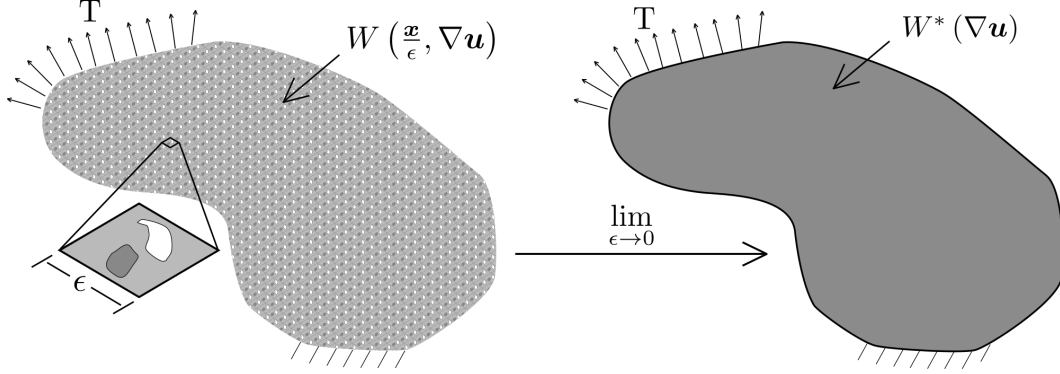


Figure 1.4: Two-scale nonlinear elasticity boundary value problem with essential boundary conditions and applied tractions \mathbf{T} . As the characteristic size of the unit cell ε tends towards zero, homogenization attempts to find a smooth homogenized energy density $W(\nabla \mathbf{u})$ to approximate the highly oscillatory energy density of the

conditions on $\partial\Omega_2$ (see Fig. 1.4). Here, W denotes the strain energy density as a function of position \mathbf{x} and the displacement gradient, \mathcal{L} is the external force potential³, and ε is a characteristic size of the RVE. That is, we have

$$\mathbf{u}^\varepsilon = \arg \min \{ \mathcal{I}[\mathbf{u}^\varepsilon] \mid \mathbf{u}^\varepsilon \in H_0^1(\Omega) \}, \quad (1.9)$$

where $H_0^1(\Omega) \subset H^1(\Omega)$ is the set of functions satisfying the Dirichlet boundary conditions.

Since the material's constitutive response fluctuates on the scale of $\mathbf{y} = \mathbf{x}/\varepsilon$ whereas the boundary value problem (i.e. the geometry and any loading) varies on the scale of \mathbf{x} , one ideally seeks to find a homogeneous energy density $W^*(\nabla \mathbf{u})$ such that Eq. (1.8) converges to the homogeneous total potential energy

$$\mathcal{I}[\mathbf{u}] = \int_{\Omega} W^*(\nabla \mathbf{u}) dV - \mathcal{L}(\mathbf{u}), \quad (1.10)$$

in some sense. In particular, it is desired that Eq. (1.8) Γ -converges to Eq. (1.10), which guarantees the convergence of minimizers of the functionals. As shown in Geymonat et al. [46], if $W(\cdot, \nabla \mathbf{u})$ is convex, the homogenized energy density can be computed by solving the RVE problem

$$W^*(\nabla \mathbf{u}) = \inf \left\{ \frac{1}{|\Omega_{\text{UC}}|} \int_{\Omega_{\text{UC}}} W(\mathbf{y}, \nabla \mathbf{u} + \nabla \mathbf{v}) dV \mid \mathbf{v} \in V(\Omega_{\text{UC}}) \right\}, \quad (1.11)$$

with $V(\Omega_{\text{UC}}) \subset H^1(\Omega_{\text{UC}})$ being the set of all periodic H^1 -functions on Ω_{UC} with zero mean, i.e. $\int_{\Omega_{\text{UC}}} \mathbf{v} dV = \mathbf{0}$.

³In the case of Fig. 1.4, the external force potential is $\mathcal{L}(\mathbf{u}^\varepsilon) = \int_{\partial\Omega_2} \mathbf{T} \cdot \mathbf{u}^\varepsilon dS$.

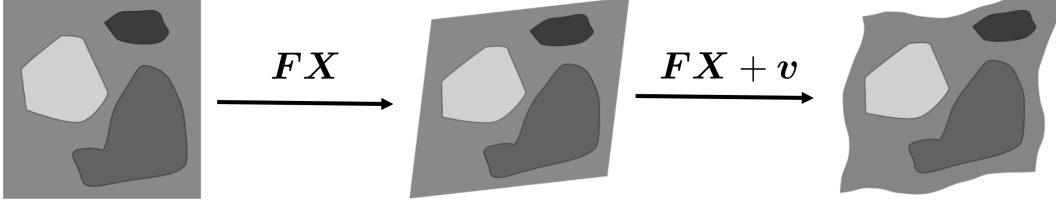


Figure 1.5: The unit cell problem with periodic boundary conditions. The average deformation gradient of the macroscale \mathbf{F} is applied to the unit cell, and the fine-scale correction \mathbf{v} is determined by minimizing the energy of the unit cell over all periodic corrections.

In the case of linear elasticity, the energy density is convex, and Eq. (1.11) holds (see Cioranescu and Donato [30] for a detailed discussion of homogenization in linear elasticity). However, for finite deformations, the energy density is in general not convex⁴, and the homogenized energy density is given by

$$W^*(\nabla \mathbf{u}) = \inf_{k \in \mathbb{Z}} \left\{ \inf \left\{ \frac{1}{|k\Omega_{\text{UC}}|} \int_{k\Omega_{\text{UC}}} W(\mathbf{y}, \nabla \mathbf{u} + \nabla \mathbf{v}) dV \mid \mathbf{v} \in V(k\Omega_{\text{UC}}) \right\} \right\}, \quad (1.12)$$

where $k\Omega_{\text{UC}}$ is a k^d sized tessellation of RVEs in d dimensions.

The single RVE problem in Eq. (1.11) is not difficult to solve, and can be thought of as affinely deforming the RVE using the deformation gradient, then allowing a periodic displacement field across the unit cell in order for the RVE to "relax" (see Fig. 1.5). On the other hand, Eq. (1.12) requires performing the same computation over an unbounded number of unit cells, which is impractical, and is one of the reasons why homogenization of solids undergoing finite deformations is so difficult.

In order to use the aforementioned homogenization techniques for linear elastic periodic truss lattices, the unit cell of the truss is used as the RVE. Then, the truss members are either discretized with continuum finite elements or beam finite elements, with the latter resulting in the discrete RVE problem

$$\begin{aligned} W(\nabla \mathbf{u}) = \inf_{\mathbf{u}, \boldsymbol{\theta}} & \left\{ \frac{1}{|\Omega_{\text{UC}}|} \sum_{e \in \mathcal{E}} W_e(\mathbf{u}, \boldsymbol{\theta}) \right\} \\ \text{s.t. } & \mathbf{u}(\mathbf{X}^+) - \mathbf{u}(\mathbf{X}^-) = \nabla \mathbf{u}(\mathbf{X}^+ - \mathbf{X}^-) \\ & \boldsymbol{\theta}(\mathbf{X}^+) - \boldsymbol{\theta}(\mathbf{X}^-) = \mathbf{0}, \end{aligned} \quad (1.13)$$

⁴In fact, a convex energy density for finite deformations violates material frame indifference.

where W_e is the energy of an individual beam in the truss, \mathbf{u} and $\boldsymbol{\theta}$ are the translational and rotational dofs of the beam finite elements, and \mathbf{X}^+ and \mathbf{X}^- represent points on opposite sides of the unit cell. Since Eq. (1.13) usually has relatively few degrees of freedom to minimize, the homogenized energy density can sometimes be computed analytically. This was done for some 2D linear elastic truss lattices, and the homogenized stiffness tensors in Voigt notation are included in Appendix A.

1.5 Properties of Periodic Truss Lattices

1.5.1 Bending- and Stretching-Dominated Lattices

The analysis of kinematically determinate truss structures was extended to the case of an infinite periodic truss lattice in Deshpande et al. [34]. Similar counting of the number of beams and nodes used in Maxwell's relation for an infinite periodic truss leads to the necessary condition that the average number of beams connected at each node must be $Z = 4$ in 2D or $Z = 6$ in 3D in order for a truss to be rigid. Note again that the connectivity of the truss is a necessary, but not sufficient condition for rigidity.

Hutchinson and Fleck [54] looked at whether periodic truss lattices were able to produce any macroscopic strain without stretching any beam by applying Bloch boundary conditions to a unit cell. Here, the same concept is described, but homogenization theory is used to characterize the periodic truss lattice.

If the periodic truss lattice is assumed to be linear elastic, the elasticity tensor of the homogenized material is given by the second derivative of the homogenized energy density given by Eq. (1.13): $\mathbb{C} = \partial^2 W^* / \partial \boldsymbol{\epsilon}^2$. Truss lattices can be grouped into two categories based on the eigenvalues of the elasticity tensor. If the minimum eigenvalue of the elasticity tensor scales with the stretching stiffness of a beam, this means there are no macroscopic strain producing deformations where the beams undergo only bending (otherwise, that deformation mode would scale like the bending stiffness of a beam, and would be the minimum eigenvalue). On the other hand, if the homogenized elasticity tensor has at least one eigenvalue that scales with the bending stiffness of a beam, then a macroscopic strain-producing deformation mode exists such that no beams are stretched. When all of the eigenvalues scale with the stretching stiffness, the periodic truss lattice is called *stretching-dominated*, and when there exists at least one eigenvalue that scales with the bending stiffness of a

beam, it is called *bending-dominated*, viz.

$$\begin{aligned} \min_{\|\boldsymbol{\varepsilon}\|=1} \boldsymbol{\varepsilon} \cdot \mathbb{C} \boldsymbol{\varepsilon} &\sim \left(\frac{r}{L}\right)^2 \iff \text{stretching-dominated} \\ \min_{\|\boldsymbol{\varepsilon}\|=1} \boldsymbol{\varepsilon} \cdot \mathbb{C} \boldsymbol{\varepsilon} &\sim \left(\frac{r}{L}\right)^4 \iff \text{bending-dominated} \end{aligned} \quad (1.14)$$

in 3D and

$$\begin{aligned} \min_{\|\boldsymbol{\varepsilon}\|=1} \boldsymbol{\varepsilon} \cdot \mathbb{C} \boldsymbol{\varepsilon} &\sim \left(\frac{r}{L}\right) \iff \text{stretching-dominated} \\ \min_{\|\boldsymbol{\varepsilon}\|=1} \boldsymbol{\varepsilon} \cdot \mathbb{C} \boldsymbol{\varepsilon} &\sim \left(\frac{r}{L}\right)^3 \iff \text{bending-dominated} \end{aligned} \quad (1.15)$$

in 2D. We will use the same nomenclature of bending- and stretching-dominated lattices to also refer to the individual strain modes, i.e. a strain mode that has a stiffness proportional to the stretching stiffness of a beam is deemed a stretching-dominated mode.

Instead of using the ratio r/L to describe the slenderness of the periodic truss lattice, the *relative density*, $\bar{\rho}$, is commonly used, which is the ratio of the volume of solid to air in the periodic truss,

$$\bar{\rho} = \frac{\sum_b V_b}{V_{\text{UC}}}, \quad (1.16)$$

where V_b is the volume of a beam, and V_{UC} is the volume of a unit cell⁵. The stiffness scaling relations for bending- and stretching-dominated unit cells are then commonly expressed as

$$E^* \sim E \bar{\rho}^b, \quad (1.17)$$

with E^* being the smallest elastic modulus of the effective material, and $b = 1$ for stretching dominated lattices, and $b = 3$ or $b = 2$ in 2D or 3D respectively.

As a review, for periodic truss lattices,

$$\begin{aligned} \text{kinematically determinate} &\iff \text{rigid} \implies \text{stretching-dominated} \\ \text{bending-dominated} &\implies \text{kinematically indeterminate} \iff \text{non-rigid}. \end{aligned} \quad (1.18)$$

Figure 1.6 gives examples of rigid, non-rigid, stretching-dominated, and bending-dominated unit cells of periodic truss lattices.

⁵Equation (1.16) assumes that the truss members are slender and neglects overlaps of the beams at truss junctions. See the supplementary materials of Meza et al. [71] for a discussion on the higher-order terms in the relative density calculations of truss lattices.

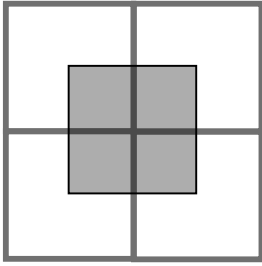
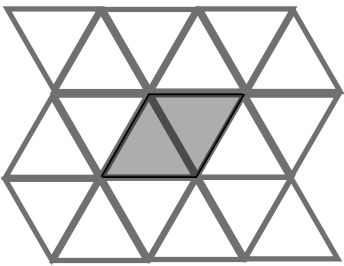
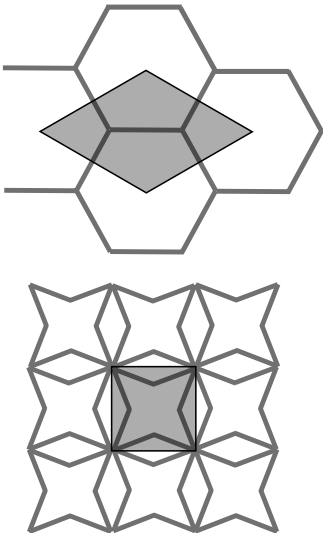
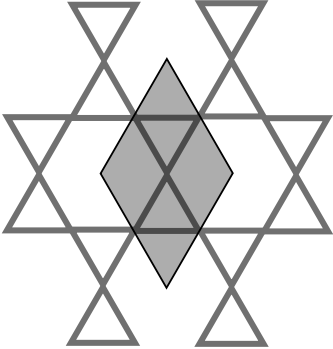
	Bending-Dominated	Stretching-Dominated
Bravais Lattice		
Multilattice		

Figure 1.6: Categorizing lattices and bending- or stretching-dominated lattices and as Bravais lattices or multilattices.

It is important to note here that the term bending-dominated seems to imply that the deformation of the periodic truss lattice is dominated by the bending of individual beams, but this is only the case for certain deformation modes. For instance, the square lattice (see Fig. 1.6) is a bending-dominated lattice since shear deformations are bending-dominated. However, axial strain in the x - or y -directions are stretching-dominated modes. Similarly, the hexagon lattice is a bending-dominated lattice but has a stretching-dominated bulk modulus, thus the periodic lattice behaves as a nearly incompressible homogenized material. The entire elastic modulus tensors of select 2D periodic truss lattices are in Appendix A.

Commonly, periodic truss lattices are also categorized by the number of nodes inside each unit cell of the lattice. In Deshpande et al. [34], lattices with a single node per unit cell are said to be *similarly situated*. However here, they are referred to as *Bravais lattices*, and truss lattices with more than one node per unit cell are called *multilattices* (see Section 3.1 for more details).

1.5.2 Effective Mechanical Properties

As the stiffness of bending- and stretching-dominated lattices scale differently with the relative density of the lattice, so do other mechanical properties. Gibson and Ashby [48] showed that the yield strength of truss lattices also follows a scaling law with relative density

$$\sigma_y^* \sim \sigma_y \bar{\rho}^c. \quad (1.19)$$

where σ_y^* is the failure stress of the homogenized material, σ_y is the failure stress of the constituent material, and $c = 1$ for stretching-dominated lattices, and $c = 2$ or $c = 1.5$ for bending-dominated lattices in 2D and 3D, respectively. As is the case with the stiffness, the scaling laws apply to specific modes of deformation, rather than the lattice as a whole.

The fracture toughness of truss lattices has also been investigated through various theoretical and computational techniques. Most of the work on the fracture toughness of truss lattices used expensive structural element simulations [1, 29, 33, 41, 82, 89, 94, 98, 100, 106, 109]. One popular method for determining the brittle fracture toughness of a lattice is the boundary layer method [94], which applies the K-field displacements to the boundary of a large notched lattice (see Fig. 1.7). On the other hand, there has been analytical work on computing the fracture toughness of lattices. Chen et al. [28] used effective continuum material models to calculate the corresponding stress fields around a crack tip, and Lipperman et al. [64] used the representative cell method of Ryvkin and Nuller [91] to analytically compute the brittle fracture toughness of infinite 2D truss lattices.

All of the work suggests the fracture toughness of a truss lattice has a power-law scaling with the relative density

$$\frac{K_{IC}^*}{\sigma_y \sqrt{L}} \sim \bar{\rho}^d, \quad (1.20)$$

where K_{IC}^* is the fracture toughness of the effective material. Unlike the stiffness and strength properties where the scaling law seems to be uniquely

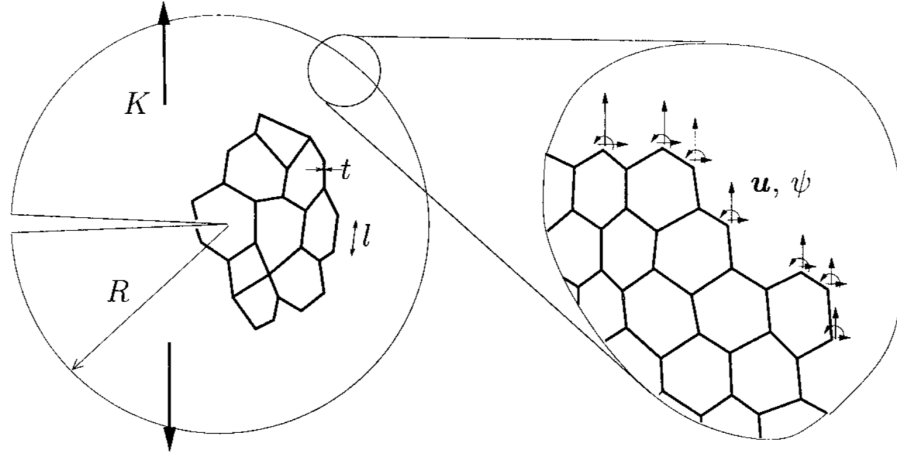


Figure 1.7: Depiction of the boundary layer method for a cellular material. Reprinted by permission from Springer Nature: International Journal of Fracture [94], copyright 2001.

determined by whether the lattice (or more specifically, deformation mode) is bending- or stretching-dominated, the fracture toughness scaling law differs even among lattices in the same class. For instance, the triangle lattice has a scaling exponent of $d = 1$, but the kagome lattice has an exponent of $d = 1/2$, and the hexagon lattice has an exponent of $d = 2$. For more details on the fracture toughness of truss lattices, see Alonso and Fleck [1].

These scaling laws convey some of the most useful properties of periodic truss lattices. For instance, a truss lattice made out of a stiff and strong bulk material (e.g. aluminum), could be used to create an effective material with a stiffness and strength proportional to the bulk properties and the relative density of the lattice. If the lattice is made with sufficiently low relative density, the metamaterial would have the same density as certain foams, but with much higher stiffness and strength due to the designed microstructure (see Fig. 1.8). These truss lattices open up a new area of the design space for lightweight materials that are also strong, stiff and tough. Furthermore, periodic truss lattices can be designed to give certain macroscopic properties (e.g. anisotropy [104]) simply by changing the topology of the unit cell. This opens up the possibility of using the same constituent material to manufacture multiple metamaterials with different properties tailored for specific applications.

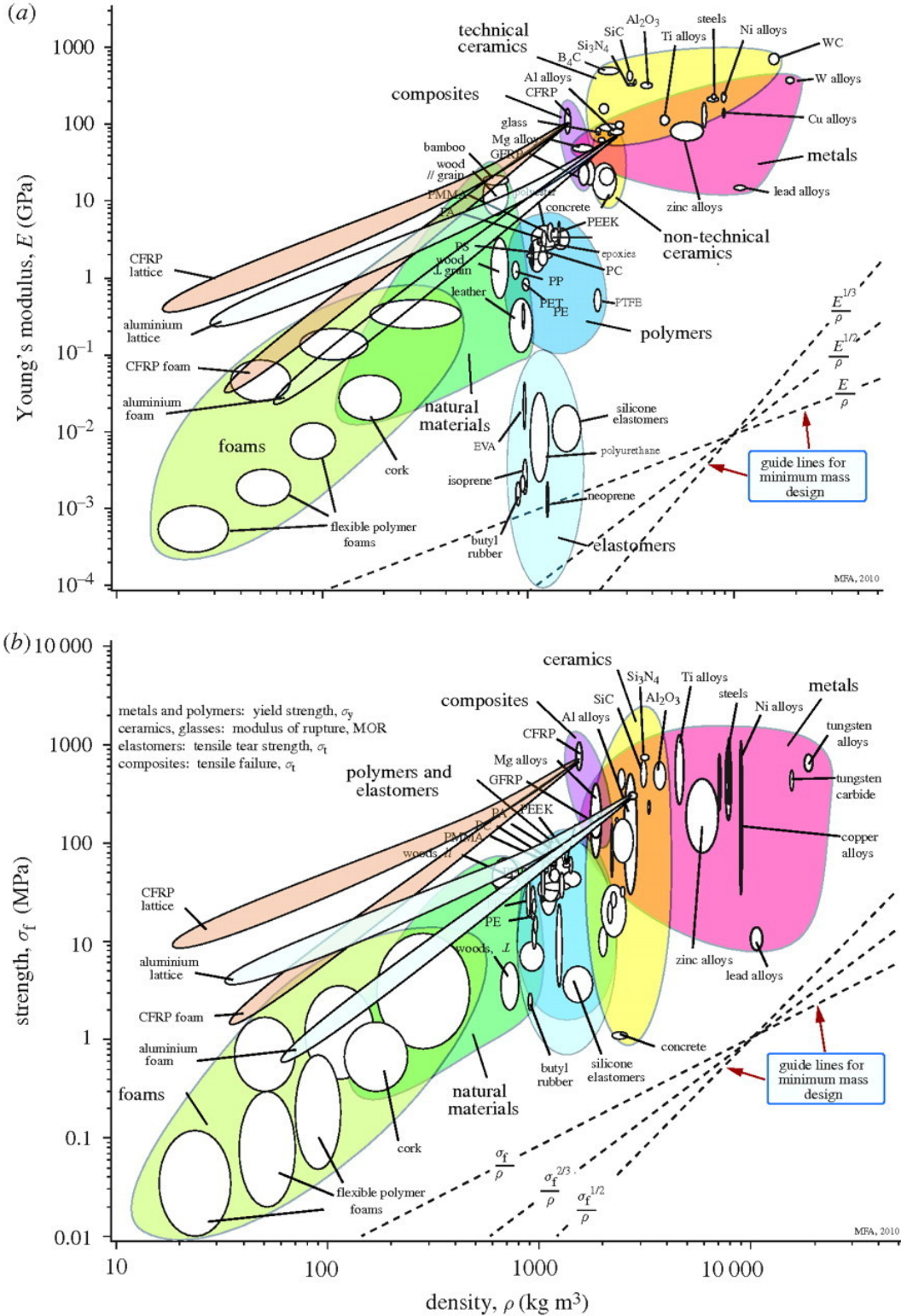


Figure 1.8: Elastic moduli and yield strength of various materials. Note that the predicted stiffness and strength of lattices made of aluminum or carbon fiber reinforced polymers (CFRP) populate otherwise empty regions of the material space. Reproduced from Fleck et al. [40] with permission of the copyright holder, The Royal Society.

NUMERICAL MODELING OF TRUSS STRUCTURES

2.1 High-Fidelity RVE Characterization

The analysis in Section 1.5 relies on the slender-beam assumption such that each truss member can be modeled as a linear elastic Euler-Bernoulli beam. However, non-slender trusses or lattices that undergo large, nonlinear deformation (e.g. buckling, plasticity, etc.) cannot be analyzed with such simple models. For this reason, some resort to using more refined microstructural models - either using nonlinear beams (e.g. [113]) or a full continuum-level finite element discretization of the unit cell problem in order to better understand the nonlinear constitutive response of the homogenized material [22, 71, 92, 103, 105]. The higher fidelity of continuum-level finite elements allows for the straightforward modeling of higher-order effects (e.g. nodes), material nonlinearity (e.g. plasticity), and allows for the accurate modeling of other unit cells where beam theory does not apply (e.g. hollow trusses, general cellular solids).

It is worth noting that if the truss structure is allowed to undergo large deformations and/or rotations, then the problem is no longer convex, and the intractable k -RVE problem Eq. (1.12) must be solved to obtain the nonlinear response of the homogeneous material. In practice, Eq. (1.12) is approximated by successively increasing k (i.e. increasing the number of unit cells in the RVE) until the homogenized energy density does not change [113], or by probing longer wavelength instabilities using Bloch wave boundary conditions [56].

Meza et al. [71] performed both continuum-level finite element simulations and simulations using structural elements of various three-dimensional truss lattices to understand the applicability of beam theory in the linear elastic regime to different truss lattices. The results showed that the stiffness of non-slender or hollow lattices can differ significantly from beam theory, but the stiffness of solid slender lattices matched those of beam theory (see Fig. 2.1).

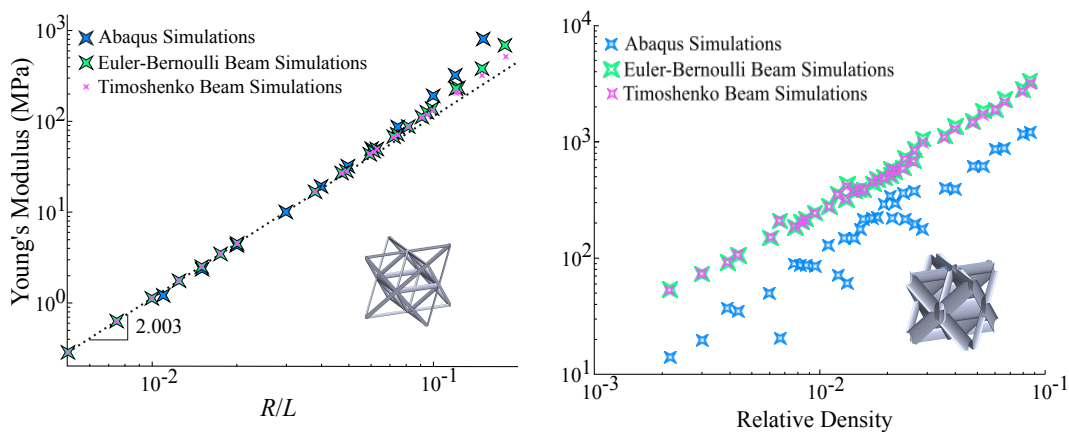


Figure 2.1: Computed homogenized stiffness of solid (left) and hollow shell (right) octet truss lattices using continuum-level finite element analysis (with the commercial code Abaqus), and structural Euler-Bernoulli and Timoshenko beams for various relative densities. Reprinted from [71].

2.2 Fully-Resolved Simulations

In order to solve boundary value problems of finite-sized truss lattices, some use fully-resolved finite element techniques where the entire truss geometry is discretized either with continuum or structural finite elements. Of course, the highest fidelity simulations are those which discretize the truss using continuum finite elements (e.g. [104], or shell elements in the case of hollow lattices [73]; see Fig. 2.2), and can be used to model geometric and material nonlinearity. However, the computational cost of these simulations limits the size of the truss that can be modeled. Recently, there has also been an effort to creating reduced-order models of nodes of non-slender truss lattices in order to capture the higher-order effects while keeping computational costs low, enabling the modeling of larger lattices [81].

Simulations that use structural finite elements (e.g. with each truss member modeled with a beam finite element) are less expensive and can be used to model larger truss lattices [1, 29, 33, 41, 53, 78, 82, 89, 94, 100, 106, 107]. Fully-resolved structural finite element models have the ability to incorporate geometric and material nonlinearities, as well as the ability to model more general non-periodic truss lattices. However, the number of structural elements that can be modeled is still limited due to the computational cost. As the sheer number of truss members contained in such truss lattices disqualifies brute-force modeling techniques due to extreme computational expenses, multiscale modeling techniques become the method of choice.

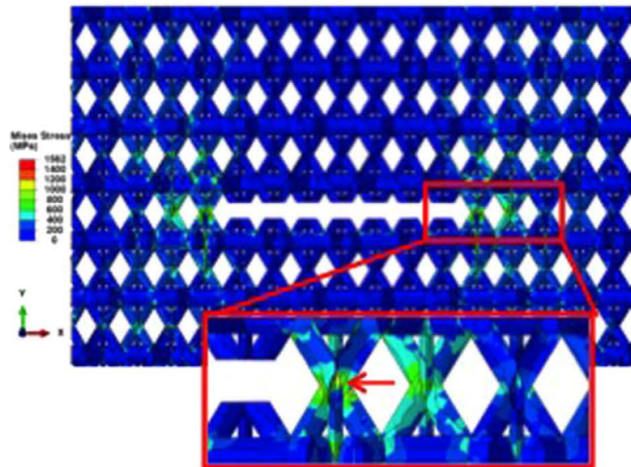


Figure 2.2: Simulation of uniaxial tension of a notched hollow octet truss lattice modeled using shell finite elements to investigate the insensitivity of truss lattices to flaws. Reprinted from [73].

2.3 Numerical Homogenization

One of the most common techniques for solving multiscale boundary value problems like Eq. (1.9) is that of numerical homogenization. It attempts to approximate solutions to the homogeneous problem in Eq. (1.10). Since finding a closed-form solution to Eq. (1.11) (or Eq. (1.12) if the energy is nonconvex) is generally not possible, the microscale RVE problem is solved numerically in order to compute $W^*(\nabla \mathbf{u})$ or its derivatives — coupling the two scales. Figure 2.3 gives an example of the numerical homogenization approach and shows the coupling between scales. Since the macroscale and microscale are treated separately, there is an inherent assumption of a separation of scales between the micro- and macro-scale (i.e. $\epsilon \ll 1$). Therefore, these techniques are referred to as a *hierarchical* multiscale techniques.

One of the most well-known numerical homogenization techniques is the FE^2 method [39, 47]. In the FE^2 method, the finite element method is used to approximate both the homogenized problem, Eq. (1.10), and the RVE problem, Eq. (1.11) (or Eq. (1.12) is approximated if the energy density is not convex). For trusses, the macroscale BVP is usually approximated with continuum finite elements, while structural finite elements are used to model the truss in the RVE [7, 76, 79, 113]. There are also some models that use a higher-order version of the RVE problem that depends on the deformation gradient, \mathbf{F} , and its gradient, thereby introducing a length scale into the problem [35, 38, 45, 61, 85, 110].

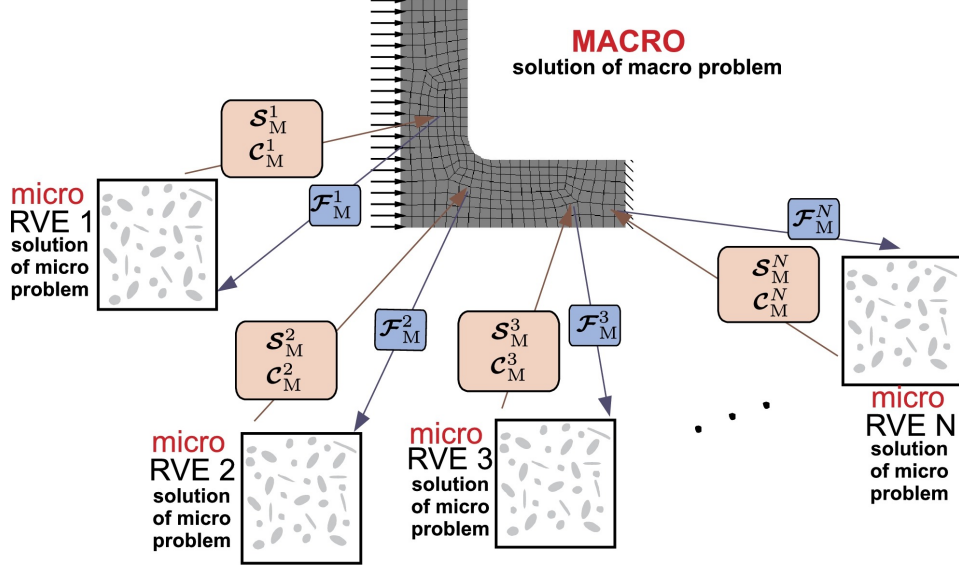


Figure 2.3: Pictorial description of numerical homogenization reprinted from Matouš et al. [67] with permission of the copyright holder, Elsevier. \mathbf{F}_M represents the macroscopic deformation gradient, and \mathbf{S}_M and \mathbf{C}_M represent the macroscopic stress and tangent matrices that are returned to the macroscale after solving the RVE problem.

Another way of thinking about these hierarchical multiscale methods is that they aim to approximate the solution of the heterogeneous fine-scale problem by looking for solutions through asymptotic expansion. For the case of first-order homogenization (i.e. where strain gradients are not included), the methods look for a solution of the form [30]

$$\mathbf{u}_\epsilon(\mathbf{x}) = \mathbf{u}(\mathbf{x}) + \epsilon \mathbf{v}\left(\mathbf{x}, \frac{\mathbf{x}}{\epsilon}\right) + O(\epsilon^2), \quad (2.1)$$

where \mathbf{u} is the smooth displacement computed from the homogenized boundary value problem, and \mathbf{v} represents the fine-scale periodic oscillations computed from the RVE problem.

One contention with these hierarchical multiscale methods like FE^2 is that the essential boundary conditions are only satisfied in an average sense. It is desired to enforce $\mathbf{u}_\epsilon(\mathbf{X}) = \mathbf{u}_0(\mathbf{X})$ on the boundary, but FE^2 only enforces $\mathbf{u}(\mathbf{X}) = \mathbf{u}_0(\mathbf{X})$. Since the average of \mathbf{v} over a unit cell is zero, the displacement field averaged over a unit cell at the boundary satisfies the essential boundary conditions rather than the fine-scale displacement field itself. In linear elasticity, however, the energy difference between the two-scale expansion Eq. (2.1) and the true solution is $\mathcal{O}(\epsilon^{1/2})$ [30]. Loosely speaking, this means

that FE^2 matches the global behavior of the body, but it does not capture boundary effects.

Similar to how the FE^2 method does not satisfy boundary conditions exactly, it also does not exactly enforce compatibility of the displacement field. For instance, take two elements K_1 and K_2 in the macro-scale finite element discretization. At the edge ∂K_{12} between the two elements, in the two-scale expansion Eq. (2.1), we have $\mathbf{u}^{K_1}(\mathbf{x}) = \mathbf{u}^{K_2}(\mathbf{x})$ (due to conformity of the shape functions on the macro-scale) but $\mathbf{v}^{K_1}(\mathbf{x}) \neq \mathbf{v}^{K_2}(\mathbf{x})$, since $\mathbf{v}(\mathbf{x})$ depends on $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ which is not necessarily the same in K_1 and K_2 . The displacement field is only conforming on the macro-scale, but not on the microscale. The displacement field is conforming at element boundaries in the same way that it satisfies boundary conditions: the displacement field averaged over a unit cell is conforming. One could make similar arguments to those made above with the boundary conditions that this lack of conformity does not incur large errors in the solution as long as the separation of scales holds.

2.4 Quasicontinuum Method

Another multiscale modeling technique that is gaining interest in the truss lattice modeling community is the Quasicontinuum (QC) method — a *concurrent* multiscale technique that does not require a separation of scales. Although originally developed to efficiently model atomic crystals [101], the QC method has recently been applied to truss lattices [15–20, 86–88]. Unlike numerical homogenization methods, the QC method starts from a finite-sized lattice, then applies kinematic constraints and energy approximation techniques (also called *summation rules* or *sampling rules*) to drastically reduce the number of degrees of freedom (dofs) and the computational costs associated with evaluating the mechanical quantities. In regions of interest (e.g. large strains, plasticity, etc.), the kinematics and the energy of the lattice are computed exactly, while in regions of less interest, the kinematic constraints and energy approximation techniques are used to approximate the behavior of the lattice.

The kinematic constraints in the QC method are developed by choosing *representative sites* (commonly called repatoms in atomistic simulations and repnodes for trusses) which retain all of their dofs, and dofs of the remaining lattice sites are interpolated from the representative sites (see Fig. 2.4). Most commonly, linear interpolation on a Delaunay-triangulated mesh is used, although

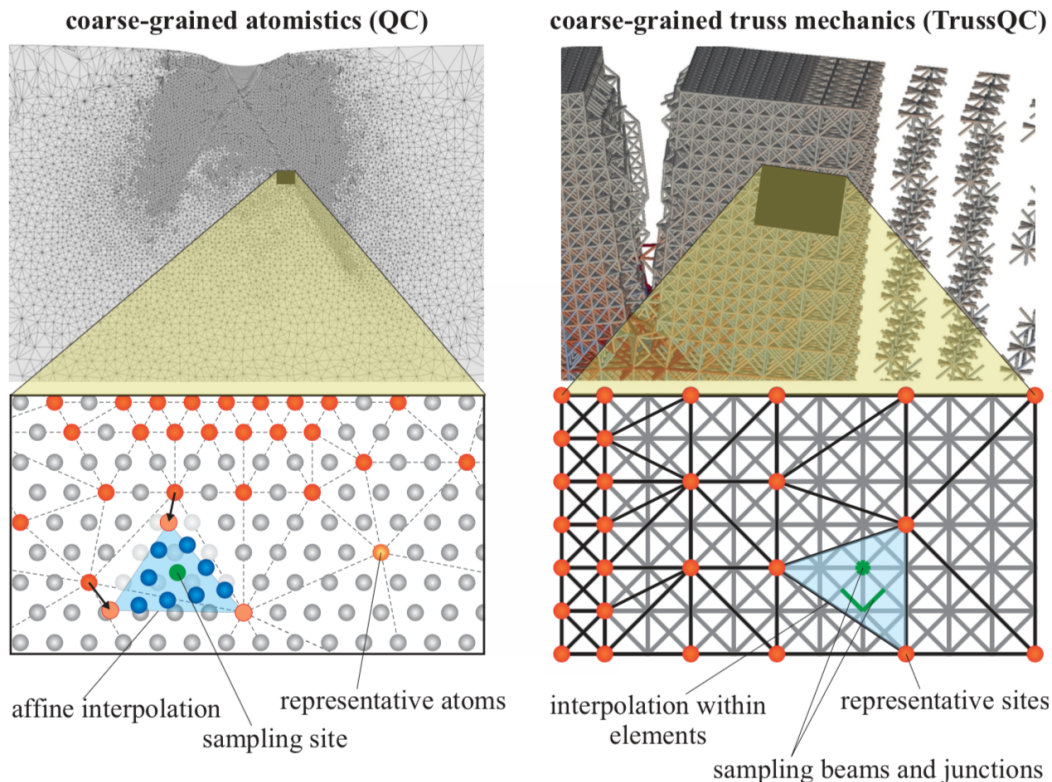


Figure 2.4: The traditional QC method for atomic lattices (left) and the extension to truss lattices (right).

there have been formulations with more complex interpolation schemes such as meshless methods [111] and higher-order interpolations [16, 114]. While there are relatively few unique interpolation schemes, there are a wide variety of QC techniques used to approximate the global energy of the lattice.

The first distinction of the QC methods comes from the so-called *local* and *nonlocal* treatment of coarse-grained regions in the simulation. Local regions of the simulation are regions where the energy of the lattice is approximated using only the deformation gradient at a given point. Usually, this is done using the Cauchy-Born rule — an assumption that the neighboring lattice sites undergo affine deformation (in addition to relative shifts of the lattices in the multilattice) defined by the deformation gradient at a point. In nonlocal regions of the simulation, no such assumption is made, and the energy of the lattice is computed exactly by using the true kinematics of the lattice.

The first version of the QC method proposed in Tadmor et al. [101] is a *local-nonlocal* version of the QC method. In fully-resolved regions, the exact non-local energy is computed, but in the coarse-grained regions, the Cauchy-Born

rule is used to approximate the energy. There are also *fully local* versions of the QC method where the Cauchy-Born rule is used everywhere [99]. However, these methods are extremely similar to the numerical homogenization methods with a single atom as the unit cell, and molecular statics (MS) is used to compute the energy of the unit cell problem (so a suitable name would be FE-MS). Lastly, there are *fully-nonlocal* versions of the QC method, where no distinction is made between the fully-resolved regions and coarse-grained regions [2, 3, 37].

Various methods have been proposed for sampling the mechanical quantities (such as energy and forces) in the coarse-grained regions including local, Cauchy-Born-type approximations [101], force- and energy-based summation rules [37], cluster sampling [58], numerical quadrature [49], and combinations of Cauchy-Born and nonlocal sampling rules [3]. It should be noted that all of these summation rules are ad-hoc, developed for a specific interpolation rule — usually linear interpolation on a Delaunay triangulated mesh — and do not naturally generalize to higher-order interpolations.

Most implementations of the QC method are intended to work specifically with Bravais lattices. That is, lattices where all atoms or nodes of a truss lie on sites in the lattice set

$$\Lambda_B = \left\{ \mathbf{X} \in \mathbb{R}^d \mid \mathbf{X} = \sum_{j=1}^d b_j \mathbf{a}_j + \mathbf{S}, \quad b_j \in \mathbb{Z} \right\}, \quad (2.2)$$

where \mathbf{a}_j are the primitive lattice vectors, and \mathbf{S} represents a rigid body shift of the entire lattice. However, there are a few instances of the QC method used for more complex atomic lattices¹ [102], which have atoms that lie on the sites in the set

$$\Lambda_M = \left\{ \mathbf{X} \in \mathbb{R}^d \mid \mathbf{X} = \sum_{j=1}^d b_j \mathbf{a}_j + \mathbf{S}^\alpha, \quad b_j \in \mathbb{Z}, \quad 1 \leq \alpha \leq n_B \right\}. \quad (2.3)$$

This set can be described as being made up of n_B Bravais lattices shifted relative to one another by the shift vectors \mathbf{S}^α . Therefore, it is common to refer to these lattices as *multilattices*. Hexagonal close packed (HCP) crystals are an example of an atomic multilattice, and the hexagon and kagome trusses are examples of 2D truss multilattices (see Fig. 3.1).

¹There has not been any QC formulation used for truss multilattices.

The first implementation of the QC method for multilattices in Tadmor et al. [102] is a local-nonlocal version of the QC method. In the fully-resolved region, each unit cell location was given extra dofs δ^α , $\alpha \in \{1, \dots, n_B\}$ representing the relative shifts of the n_B additional atoms at each unit cell site (the shift of the first Bravais lattice was defined to be zero, $\delta^1 = \mathbf{0}$). However, in the coarse-grained region, sites only contained d dofs in d dimensions, and the atomistic equivalent of the RVE problem, Eq. (1.11), with a single unit cell was solved in each coarse-grained element. Since the atomic interactions are non-convex, the method was improved to approximate the atomistic equivalent of the k-RVE problem, Eq. (1.12), using a method called cascading Cauchy-Born kinematics [36]. Cascading Cauchy-Born kinematics uses Bloch wave techniques to probe instabilities with wavelengths longer than a single unit cell, and allows more unit cells to be incorporated into the RVE. This allows the local-nonlocal QC method to capture phase transformations in the coarse-grained regions.

The extension of the traditional atomistic QC method to model truss lattices is rather straightforward: nodes of the truss replace atoms, and finite element beam or bar energies are used in place of inter-atomic potentials (see Fig. 2.4). Initial studies assumed the truss members to behave as linear elastic springs; later extensions included inelasticity and failure of truss members [18, 19, 59, 86] as well as introducing rotational dofs at each node and using Euler-Bernoulli beam connections to model the stretching and bending of truss members [16]. Concepts were further paired with the eXtended Finite Element Method (XFEM) to track the propagation of cracks in the lattice (see Fig. 2.5) [88]. However, all of these implementations have been limited to local-nonlocal versions of the QC method only for trusses that can be defined by a 2D Bravais lattice (either square or cross-braced lattices).

2.5 Shortcoming of Existing Methods

While there has been an extensive amount of work devoted to numerical modeling of truss structures, there is still a gap in the modeling capabilities.

Fully-resolved continuum finite element models are the highest fidelity models — capable of capturing strong material nonlinearity and not requiring any assumption on the slenderness of truss members — but are extremely expensive to solve, limiting the size of the simulation [73, 104]. Fully-resolved structural finite element models (i.e. beam elements) use the slender beam assumption

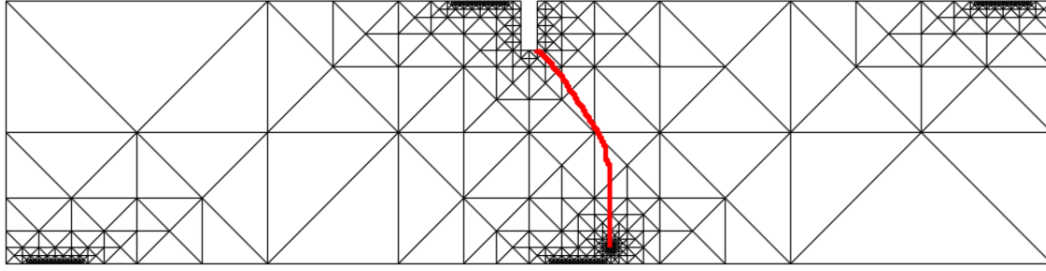


Figure 2.5: The QC method paired with aspects of the extended finite element method to model the crack propagation (crack path shown in red) in a cross-braced truss lattice. Reprinted from Rokoš et al. [88] with permission from the copyright holder, Elsevier

to reduce the number of degrees of freedom needed to model finite-sized truss lattices. By using sophisticated beam elements, it is even possible to model geometric as well as material nonlinearity, and non-periodic truss lattices. However, even though structural finite elements are inexpensive compared to continuum-level finite element descriptions, modeling multiscale metamaterials using structural elements is not a scalable approach.

Modeling representative volume elements using either high-fidelity continuum finite elements or less expensive structural finite elements is useful in characterizing the properties of the effective material. However, whether the RVE computations are done "offline" to characterize the homogenized energy density or "online" in the FE^2 method to solve boundary value problems, there are problems with strong nonlinearities (stemming from the intractability of the k-RVE problem Eq. (1.12)). Furthermore, the homogenization procedure completely fails when there are shear-band instabilities at the microscale because there is no length scale to regularize the solution. Homogenization strategies such as FE^2 also have the additional issue of not satisfying boundary conditions exactly and therefore do not accurately capture boundary effects.

The QC method lies in a middle ground between fully-resolved models and hierarchical multiscale models. Since it starts with the fully-resolved model, the QC method is able to capture significant nonlinearities (buckling, shear bands, failure of individual trusses, etc.) and boundary effects in certain fully-resolved regions, while the coarse-graining techniques enable the method to scale up to truly multiscale lattices. The initial work in using the QC method to model trusses is promising, but so far it has only been used to model simple Bravais lattices in 2D (e.g. square, cross-braced lattices) with linearized

kinematics.

The remainder of this thesis is dedicated to filling the gaps in existing modeling methods by developing a formulation of the QC method to accurately model the *nonlinear* response of general truss *multilattices* in *three dimensions*. The novel *fully-nonlocal* multilattice formulation allows for general truss models to be used, and will be paired with *geometrically nonlinear* beam descriptions to model buckling of the truss without any separation of scales assumptions or solving microscale RVE problems. The method will be implemented with an *adaptive refinement* algorithm used to automatically improve the accuracy in regions of interest as the simulation progresses.

FUNDEMANNTALS OF THE QUASICONTINUUM METHOD

A significant portion of this chapter is taken from our recent publication:

Gregory P. Phlipot and Dennis M. Kochmann. A quasicontinuum theory for the nonlinear mechanical response of general periodic truss lattices. *Journal of the Mechanics and Physics of Solids*, 124:758 – 780, 2019. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2018.11.014>. URL <http://www.sciencedirect.com/science/article/pii/S0022509618304071>.

3.1 Initial Problem Statement

In this section, we outline the fully-resolved problem that will later be approximated with the QC method. The purpose of this section is to give definitions of unit cells, truss elements, nodes, and the total potential energy of the system which will be used in the rest of the thesis.

Periodic truss lattices, i.e. cellular networks composed of beams and beam junctions (referred to as *nodes* in the following), are produced by periodically repeating a *unit cell* along directions $\mathcal{A} = \{\mathbf{a}_1, \dots, \mathbf{a}_d\}$ with basis vectors $\mathbf{a} \in \mathbb{R}^d$ in d dimensions. A lattice Ω is defined through the set of nodes, \mathcal{P} , and the set of *truss members*¹, \mathcal{E} , i.e. $\Omega = \{\mathcal{P}, \mathcal{E}\}$. For convenience, the body of interest is decomposed into a set of unit cells. A unit cell at a location \mathbf{X}_u (in the undeformed configuration) can be defined by the set of points, \mathcal{P}_u , that lie inside the volume spanned by vectors \mathcal{A} and centered at \mathbf{X}_u (see Fig. 3.1a), i.e.

$$\mathcal{P}_u = \left\{ \mathbf{X} \in \mathcal{P} \mid \mathbf{X} = \sum_{i=1}^d \mathbf{X}_u + t_i \mathbf{a}_i, \quad t_i \in \left(-\frac{1}{2}, \frac{1}{2} \right) \text{ for } i = 1, \dots, d \right\}. \quad (3.1)$$

The unit cell also contains the set of truss members that connect the points in \mathcal{P}_u to each other, \mathcal{E}_u , and the set of truss members interacting with points in neighboring unit cells, \mathcal{E}_u^n .

For some truss lattices, (e.g. square, triangle) each unit cell consists of only a single node with connections only to nodes in neighboring unit cells (i.e.

¹We use the term truss members instead of bars or beams such that it can be used regardless of the type of structural element used to model it.

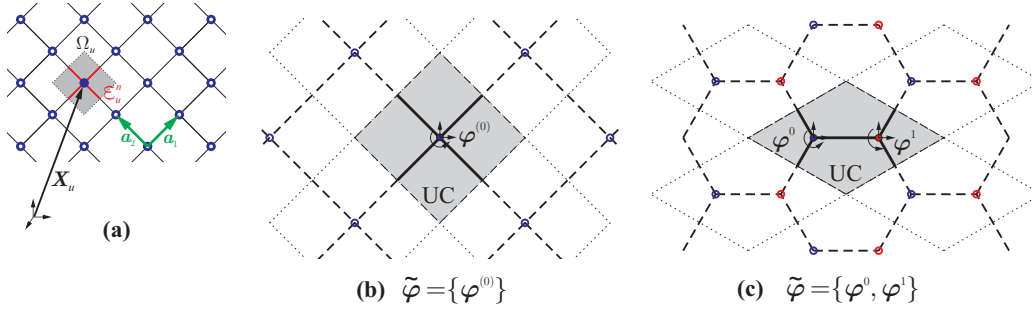


Figure 3.1: **(a)** Example definition of a unit cell Ω_u based on its node(s) (solid circle) and its connecting beams \mathcal{E}_u^n (in this case $\mathcal{E}_u = \emptyset$). Further shown are examples of **(b)** a simple Bravais lattice (diamond) and **(c)** a multilattice (hexagon, composed of two simple Bravais lattices) and their corresponding unit cells and unit cell dofs. The dofs of all nodes inside each unit cell are retained with φ denoting the generalized dofs of a truss node.

$\mathcal{E}_u = \{\emptyset\}$), as shown in Fig. 3.1(a,b) for the diamond lattice. In this case, the set of nodes lie on a Bravais lattice, with the basis vectors \mathcal{A} as the Bravais lattice vectors. That is, the locations of the nodes lie in the set described in Eq. (2.2).

More complex lattices (e.g. kagome or hexagon) are created with unit cells consisting of multiple distinct nodes (see the hexagonal lattice in Fig. 3.1c). By adopting terminology from atomistics, unit cells with multiple unique nodes are referred to as multilattices, whose set of nodes can be described by n_B Bravais lattices, each defined by the same basis \mathcal{A} but shifted relative to one another by shift vectors \mathbf{S}^α ($\alpha = 1, \dots, n_B$). Consequently, the complete point set of a multilattice is given by Eq. (2.3) and copied here for clarity:

$$\Lambda_M = \left\{ \mathbf{X} \in \mathbb{R}^d \mid \mathbf{X} = \sum_{j=1}^d b_j \mathbf{a}_j^\alpha + \mathbf{S}^\alpha, \quad b_j \in \mathbb{Z}, \quad 1 \leq \alpha \leq n_B \right\}. \quad (3.2)$$

The integer vector $\mathbf{b} = \{b_1, \dots, b_d\}$ represents the unit cell with n_B points in terms of the Bravais lattice vectors, and will hereby be referred to as the *Bravais coordinates* of the unit cell.

In some cases, it is useful to talk about the unit cells in terms of their Bravais coordinates rather than their cartesian coordinates. The mapping of the unit cells from cartesian coordinates to the Bravais coordinates is written as

$$\mathbf{b} = \mathbf{A}^{-1}(\mathbf{X} - \mathbf{S}^1), \quad (3.3)$$

where $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_d]$ is the matrix of Bravais vectors, and \mathbf{S}^1 is the offset of the lattice from the origin.

As the truss network undergoes finite deformations, we track the generalized deformed position of each node (which could include both translations and rotations) by a mapping $\boldsymbol{\varphi}(\mathbf{X})$. Rotational dofs $\boldsymbol{\theta}$ are required along with translational dofs \mathbf{x} for beam lattices subjected to bending, and from various available options we choose rotation vectors parametrized by $2d - 3$ variables (described in detail in Section 3.2.2). We denote all generalized dofs of a unit cell located at \mathbf{X}_i by $\tilde{\boldsymbol{\varphi}}(\mathbf{X}_i) = \{\boldsymbol{\varphi}^0(\mathbf{X}_i), \dots, \boldsymbol{\varphi}^{n_B}(\mathbf{X}_i)\}$; see Fig. 3.1. For convenience, $\tilde{\boldsymbol{\varphi}}^j(\mathbf{X}_i)$ will be used to denote the generalized dofs $\boldsymbol{\varphi}$ of the j^{th} point in the unit cell located at \mathbf{X}_i .

Assuming a variational constitutive model, the strain energy of a truss member $t \in \mathcal{E}$ connecting the two nodes initially at \mathbf{X}_i and \mathbf{X}_j is assumed to be given by $W_t(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j)$ and the total potential energy of a finite-size truss network follows as

$$\mathcal{I} = \sum_{t \in \mathcal{E}} W_t - \mathcal{L}, \quad (3.4)$$

with \mathcal{L} being the external force potential. It is worth noting here that by assuming that the total strain energy of the lattice can be expressed as a sum over all of the truss members in the lattice, higher-order effects in nonslender lattices (such as nodal effects) will not be captured by this model. Therefore, each truss member is modeled using a slender beam assumption, and the model is not valid for nonslender lattices or hollow lattices (as is the case with all other structural finite element simulations of truss lattices). Using the decomposition of the domain into unit cells as described above, the energy can be rewritten as a sum over the unit cells

$$\mathcal{I}(\tilde{\boldsymbol{\varphi}}) = \sum_{\Omega_u \in \Omega} W_u(\tilde{\boldsymbol{\varphi}}_u, \tilde{\boldsymbol{\varphi}}_{\mathcal{N}(u)}) - \mathcal{L}(\tilde{\boldsymbol{\varphi}}), \quad W_u = \sum_{t \in \mathcal{E}_u} W_t + \frac{1}{2} \sum_{t \in \mathcal{E}_u^n} W_t, \quad (3.5)$$

where W_u is the total energy of the unit cell (composed of energy of truss members \mathcal{E}_u fully within Ω_u and of truss members \mathcal{E}_u^n connecting to neighboring unit cells, the latter weighted by 1/2 to avoid double-counting), and $\tilde{\boldsymbol{\varphi}}_{\mathcal{N}(u)}$ are the generalized dofs of the neighboring unit cells. Finally, the solution of a quasistatic BVP is obtained from minimizing \mathcal{I} subject to essential boundary conditions:

$$\inf_{\tilde{\boldsymbol{\varphi}} \in V} \mathcal{I}(\tilde{\boldsymbol{\varphi}}), \quad (3.6)$$

where V is the set of unit cell dofs such that they satisfy the boundary conditions $\tilde{\boldsymbol{\varphi}} = \tilde{\boldsymbol{\varphi}}_0$ on $\partial\Omega$.

3.2 Modeling Individual Truss Members

In the definition of the total potential energy of the system, it was assumed that the beam could be modeled by having some energy W_t that depends on the generalized dofs at the endpoints, $\boldsymbol{\varphi}$. Here, we give some examples of what energies can be used to model the truss members, namely bars and beams with both material and geometric nonlinearities.

3.2.1 Bar Models

If the truss being modeled is a slender rigid truss lattice, then it may be desirable to neglect bending of the beams, and model the truss members as pin-jointed bars, reducing the number of dofs of the system. In this case, each node of the truss has only translational dofs $\boldsymbol{\varphi} = \{\mathbf{x}\}$, and the elastic part of the energy can be described by a one-dimensional material model

$$\frac{W_t^{\text{bar}}(\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2)}{V_b} = W(\varepsilon_{\text{ax}}), \quad (3.7)$$

where V_b is the volume of the bar, ε_{ax} is the axial strain of the truss member, $W(\varepsilon_{\text{ax}})$ is the 1D material model, and $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$ are the dofs of the nodes of the bar. For trusses undergoing infinitesimal displacements, linearized kinematics can be used, or if finite deformations are modeled, the nonlinear relation between strain and displacements can be used

$$\varepsilon_{\text{ax}}^{\text{lin}} = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot \hat{\mathbf{X}}_{\text{ax}}}{L} - 1, \quad \varepsilon_{\text{ax}}^{\text{nl}} = \frac{\|\mathbf{x}_2 - \mathbf{x}_1\|}{L} - 1, \quad (3.8)$$

where $\varepsilon_{\text{ax}}^{\text{lin}}$ and $\varepsilon_{\text{ax}}^{\text{nl}}$ are the strains under linearized kinematics and nonlinear kinematics respectively, \mathbf{x}_i are the deformed positions of the nodes of the bar, $\hat{\mathbf{X}}_{\text{ax}}$ is a unit vector point along the axis of the bar, and L is the undeformed length of the bar.

The forces and stiffness matrix of the bar follow from the definition of the energy:

$$\mathbf{F} = V_b \mathbf{B}^T \frac{\partial W}{\partial \varepsilon_{\text{ax}}}, \quad \mathbf{K} = V_b \left(\mathbf{B}^T \frac{\partial^2 W}{(\partial \varepsilon_{\text{ax}})^2} \mathbf{B} + \frac{\partial W}{\partial \varepsilon_{\text{ax}}} \mathbf{C} \right), \quad (3.9)$$

with

$$B_i = \frac{\partial \varepsilon_{\text{ax}}}{\partial \varphi_i}, \quad C_{ij} = \frac{\partial^2 \varepsilon_{\text{ax}}}{\partial \varphi_i \partial \varphi_j}. \quad (3.10)$$

The second term in the expression of the stiffness matrix is commonly referred to as the geometric stiffness, and comes about because of the nonlinear relation between strain and displacements. For the linearized kinematic bar, $\mathbf{C} = \mathbf{0}$.

This simple model for the truss members can be useful for creating inexpensive simulations of stretching-dominated trusses. However, bar elements clearly fail to model any non-rigid or bending-dominated lattice since there is no bending stiffness in the pin-jointed model. In order to model bending-dominated lattices (or have higher-fidelity models of stretching-dominated lattices), beam finite elements must be used to model the individual members.

3.2.2 Beam Models

The simplest beam model is the Euler-Bernoulli beam with Hermite interpolation. In this case, each node of the truss has translational and rotational dofs $\boldsymbol{\varphi} = \{\mathbf{x}, \boldsymbol{\theta}\}$, where $\boldsymbol{\theta} \in \mathbb{R}$ in 2D or $\boldsymbol{\theta} \in \mathbb{R}^3$ in 3D. Due to the cubic interpolation of displacements, the beam element captures the exact displacement of the beam under tip loading, which suffices for modeling truss lattices, which rarely, if ever, have loading along the length of the truss members.

The energy of an Euler-Bernoulli beam can be described in terms of the local strain measures

$$\boldsymbol{\varepsilon}^{loc} = \begin{bmatrix} \varepsilon_{ax} \\ \tilde{\boldsymbol{\theta}}^{loc} \end{bmatrix}, \quad (3.11)$$

where ε_{ax} is the axial strain of the beam, and $\tilde{\boldsymbol{\theta}}^{loc} \in \mathbb{R}^2$ in 2D and $\tilde{\boldsymbol{\theta}}^{loc} \in \mathbb{R}^6$ in 3D are the local angles of rotation per unit length of the beam around the nodes². The expressions for the local strains as a function of the node dofs $\boldsymbol{\varphi}$ depend on the beam model being used (examples are given in following sections).

The energy of an elastic Euler-Bernoulli beam can then be written as a function of the local strain values

$$W_t(\boldsymbol{\varphi}_2, \boldsymbol{\varphi}_1) = W_b(\varepsilon_{ax}, \tilde{\boldsymbol{\theta}}^{loc}). \quad (3.12)$$

²In three dimensions, all 6 local rotations at the nodes do not need to be kept as local strains since the energy of the beam only depends on the difference between the twist angles at each node. Therefore, it may be more fundamental to formulate beam energies in terms of 5 independent local angles, but this is not done here for ease of notation

The standard linear elastic beam energy density in terms of the local strains can be written as

$$W_b(\varepsilon_{\text{ax}}, \tilde{\boldsymbol{\theta}}^{\text{loc}}) = \frac{L}{2} \left(EA\varepsilon_{\text{ax}}^2 + \left(\tilde{\boldsymbol{\theta}}^{\text{loc}} \right)^T \mathbf{D} \tilde{\boldsymbol{\theta}}^{\text{loc}} \right), \quad (3.13)$$

where

$$\mathbf{D}_{2\text{D}} = \begin{bmatrix} 4EI_2 & 2EI_2 \\ 2EI_2 & 4EI_2 \end{bmatrix}, \quad \mathbf{D}_{3\text{D}} = \begin{bmatrix} GJ & 0 & 0 & -GJ & 0 & 0 \\ 0 & 4EI_2 & 0 & 0 & 2EI_2 & 0 \\ 0 & 0 & 4EI_3 & 0 & 0 & 2EI_3 \\ -GJ & 0 & 0 & GJ & 0 & 0 \\ 0 & 2EI_2 & 0 & 0 & 4EI_2 & 0 \\ 0 & 0 & 2EI_3 & 0 & 0 & 4EI_3 \end{bmatrix} \quad (3.14)$$

are the stiffness matrices for the rotational dofs in 2D and 3D, respectively, with E and G being Young's and the shear modulus, L is the undeformed length of the beam, and J , I_2 and I_3 are the polar and area moments of inertia.

Euler-Bernoulli Beam with Linearized Kinematics

Since the energies given above are in terms of the local rotations around each node and the axial strain, the beam kinematics are needed in order to compute the local strain measures from the global dofs. The simplest kinematics for this is the standard linearized kinematics. If the undeformed configuration of the Euler-Bernoulli beam is given by the rotation matrix \mathbf{R} , then the local displacements are given by

$$\mathbf{U}^{\text{loc}} = \mathbf{R}^T \mathbf{u} \quad (3.15)$$

with $\mathbf{U}^{\text{loc}} = \{u^{\text{loc}}, v^{\text{loc}}\}$ or $\mathbf{U}^{\text{loc}} = \{u^{\text{loc}}, v^{\text{loc}}, w^{\text{loc}}\}$ in 2D or 3D, respectively, being the axial and transverse displacements. Using these local displacements, the local strain measures can be written as

$$\boldsymbol{\varepsilon}^{\text{loc}} = \begin{bmatrix} \varepsilon_{\text{ax}} \\ \tilde{\theta}_1^{\text{loc}} \\ \tilde{\theta}_2^{\text{loc}} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} u_2^{\text{loc}} - u_1^{\text{loc}} \\ \theta_1 - (v_2^{\text{loc}} - v_1^{\text{loc}})/L \\ \theta_2 - (v_2^{\text{loc}} - v_1^{\text{loc}})/L \end{bmatrix} \quad (3.16)$$

in 2D and

$$\boldsymbol{\varepsilon}^{loc} = \begin{bmatrix} \varepsilon_{ax} \\ \tilde{\alpha}_1^{loc} \\ \tilde{\phi}_1^{loc} \\ \tilde{\psi}_1^{loc} \\ \tilde{\alpha}_2^{loc} \\ \tilde{\phi}_2^{loc} \\ \tilde{\psi}_2^{loc} \end{bmatrix} = \frac{1}{L} \begin{bmatrix} u_2^{loc} - u_1^{loc} \\ \alpha_1/L \\ \phi_1 - (w_2^{loc} - w_1^{loc})/L \\ \psi_1 - (v_2^{loc} - v_1^{loc})/L \\ \alpha_2/L \\ \phi_2 - (w_2^{loc} - w_1^{loc})/L \\ \psi_2 - (v_2^{loc} - v_1^{loc})/L \end{bmatrix} \quad (3.17)$$

in 3D, where $\tilde{\boldsymbol{\theta}}^{loc} = \{\tilde{\alpha}^{loc}, \tilde{\phi}^{loc}, \tilde{\psi}^{loc}\}$ and $\boldsymbol{\theta} = \{\alpha, \phi, \psi\}$ are the local and global rotations respectively.

Given the expressions for the local strain measure, the element energy, force, and stiffness matrices follow:

$$W = W_b(\boldsymbol{\varepsilon}^{loc}), \quad \mathbf{F} = \mathbf{B}^T \frac{\partial W}{\partial \boldsymbol{\varepsilon}^{loc}}, \quad \mathbf{K} = \mathbf{B}^T \frac{\partial^2 W}{(\partial \boldsymbol{\varepsilon}^{loc})^2} \mathbf{B}. \quad (3.18)$$

with

$$B_{ij} = \frac{\partial \varepsilon_i^{loc}}{\partial \varphi_j}. \quad (3.19)$$

Note that for these linearized kinematics, there is no additional term in the stiffness matrix since $\partial^2 \boldsymbol{\varepsilon}^{loc} / \partial \boldsymbol{\varphi}^2 = \mathbf{0}$.

Corotational Beam

In order to model individual truss members undergoing finite rotations while sustaining axial and bending loads, a corotational beam may be used. Here, a corotational beam formulation based on Crisfield [31], is outlined. For a detailed description of the beam elements used, the reader is referred to Crisfield [31, 32]. Dealing with finite rotations in 2D is relatively straightforward: a single rotation variable θ can be used to represent the rotation. However, in 3D there are many choices for parametrizing rotations including rotation matrices, quaternions and rotation vectors. For this corotational formulation, we consider the rotation vector $\boldsymbol{\theta} = \theta \hat{\mathbf{e}}$ which corresponds to a rotation with magnitude $\theta = |\boldsymbol{\theta}|$ around the axis defined by the unit vector $\hat{\mathbf{e}}$. It can be converted to a rotation matrix $\mathbf{R}(\boldsymbol{\theta})$ through an exponential mapping using Rodrigues' formula (for $\theta \neq 0$)

$$\mathbf{R}(\boldsymbol{\theta}) = \exp(\mathbf{S}(\boldsymbol{\theta})) = \mathbf{I} + \frac{\sin \theta}{\theta} \mathbf{S}(\boldsymbol{\theta}) + \frac{1 - \cos \theta}{\theta^2} \mathbf{S}(\boldsymbol{\theta})^2, \quad (3.20)$$

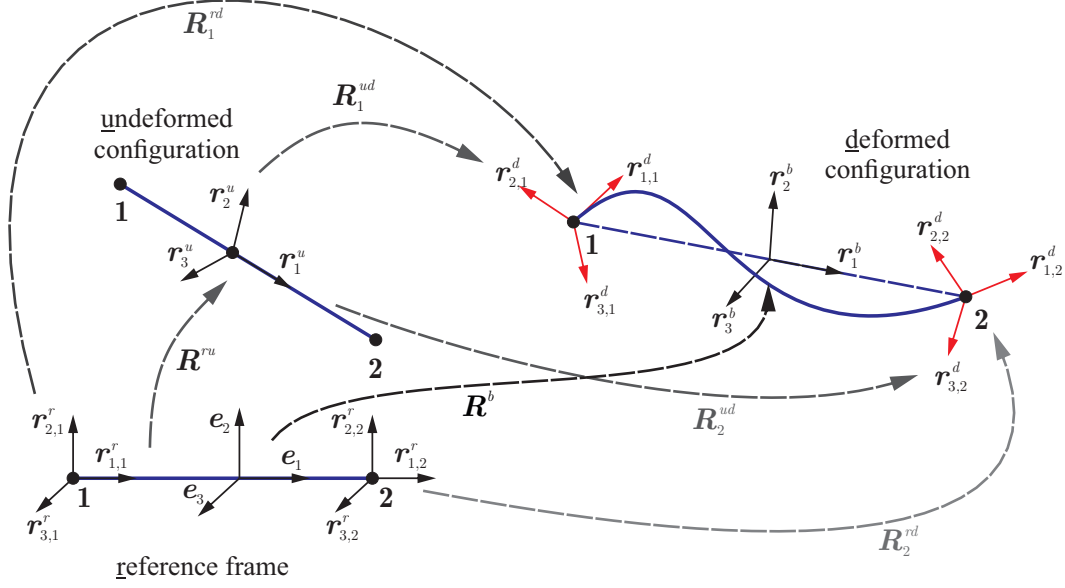


Figure 3.2: Kinematics of the 3D corotational beam: the linear Euler Bernoulli beam rotates with the element frame \mathbf{R}^b , and has small angle deflections at each node. Superscripts r , u , d , and b stand for the (r)eferece, (u)ndeformed, (d)eformed, and (b)eam configuration, respectively.

where $\mathbf{S}(\boldsymbol{\theta})$ is the skew-symmetric matrix ($\theta_i = \theta \hat{e}_i$ denoting the i th Cartesian component of $\boldsymbol{\theta}$)

$$\mathbf{S}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}. \quad (3.21)$$

By inversion of Eq. (3.20), one arrives at the logarithmic mapping

$$\boldsymbol{\theta} = \log(\mathbf{R}(\boldsymbol{\theta})). \quad (3.22)$$

The 2D and 3D corotational beam description of Crisfield [31] is based on an element frame that continuously rotates with the underlying beam element. The nonlinearity introduced in the element comes solely from the finite rotation of the element frame, while the underlying linear Euler-Bernoulli model assumes small deflections with respect to the rotating frame (see Fig. 3.2 for all definitions used in the following).

The kinematics of the corotational beams is described using three different configurations of the beam: the reference, undeformed, and deformed configurations (see Fig. 3.2). The reference configuration is taken as an undeformed Euler-Bernoulli beam element that is oriented along the positive x -axis in the

global coordinate system. The undeformed configuration represents the beam with its orientation at the start of the simulation, and can be defined by the rotation \mathbf{R}^{ru} , where the superscript ru signifies the rotation from the (r)eference configuration to the (u)ndeformed configuration. The deformed configuration represents the beam subjected to arbitrarily large displacements and rotations at the current step in the simulation. The deformed configuration is defined by the deformed nodal positions \mathbf{x}_i and global rotations \mathbf{R}_i^{rd} of each node i , where the superscript signifies the rotation from the (r)eference configuration to the (d)eformed configuration. The global rotations can also be represented by the rotation from the undeformed configuration to the deformed configuration \mathbf{R}_i^{ud} which is related to the other rotation matrices through

$$\mathbf{R}_i^{rd} = \mathbf{R}_i^{ud} \mathbf{R}^{ru}. \quad (3.23)$$

Using the global positions and rotations of each node, the beam element's orientation $\mathbf{R}^b = [\mathbf{r}_1^b, \dots, \mathbf{r}_d^b]$ is defined as an average rotation of the beam. In both 2D and 3D, \mathbf{r}_1^b is defined as the unit vector pointing from node 1 to node 2, i.e.

$$\mathbf{r}_1^b = \frac{\mathbf{x}_2 - \mathbf{x}_1}{|\mathbf{x}_2 - \mathbf{x}_1|}. \quad (3.24)$$

For the 2D corotational beam, the second axis \mathbf{r}_2^b follows from a rotation of \mathbf{r}_1^b by $\pi/2$. For the 3D corotational beam, the second and third axes are defined with the help of the average rotation matrix

$$\mathbf{R}^{av} = \mathbf{R}(\boldsymbol{\theta}^{av}) \mathbf{R}_1^{ud} \quad (3.25)$$

with $\boldsymbol{\theta}^{av}$ defined as

$$\boldsymbol{\theta}^{av} = \frac{\log(\mathbf{R}_2^{rd}) - \log(\mathbf{R}_1^{rd})}{2}. \quad (3.26)$$

With the column representation $\mathbf{R}^{av} = [\mathbf{r}_1^{av}, \mathbf{r}_2^{av}, \mathbf{r}_3^{av}]$, the remaining two axes of the beam coordinate system are defined as

$$\mathbf{r}_2^b = \mathbf{r}_2^{av} - \frac{\mathbf{r}_2^{av} \cdot \mathbf{r}_1^b}{2}(\mathbf{r}_1^b + \mathbf{r}_1^{av}), \quad \mathbf{r}_3^b = \mathbf{r}_3^{av} - \frac{\mathbf{r}_3^{av} \cdot \mathbf{r}_1^b}{2}(\mathbf{r}_1^b + \mathbf{r}_1^{av}). \quad (3.27)$$

This definition of a beam element's frame results in a coordinate system that is not necessarily orthogonal, but the deviation from orthogonality is small and therefore generally neglected (see Crisfield [31] for a discussion). Lastly, we use the global rotation at each node and the beam coordinate system to define the local (assumed to be small) rotations. This is done by realizing that

the global rotations of each node can be described by a small local rotation around the undeformed configuration $\mathbf{R}_i^{\text{loc}}$, followed by the large rotation of the beam element (i.e. $\mathbf{R}_i^{rd} = \mathbf{R}^b \mathbf{R}_i^{\text{loc}}$). This gives the definition of the local rotation matrices and the corresponding local rotation vectors as, respectively,

$$\mathbf{R}_i^{\text{loc}} = (\mathbf{R}^b)^T \mathbf{R}_i^{rd}, \quad \boldsymbol{\theta}_i^{\text{loc}} = \log(\mathbf{R}_i^{\text{loc}}). \quad (3.28)$$

To capture the global rotations of each node, the rotation \mathbf{R}^{ru} is stored in each element, so only the rotation from the undeformed to the deformed configuration must be parametrized. In 2D, this is accomplished by storing the scalar rotation angle θ ; in 3D, we choose the rotation vector $\boldsymbol{\theta}^{ud}$. It should be noted that the interpolation of our QC method calls for caution when choosing the parametrization (e.g. the linear interpolation of rotation matrices is not permissible as it does not necessarily result in a rotation matrix).

Since this representation describes the large global rotations in terms of small local rotations composed with a rigid body rotation, the energy of a corotational beam element is given by the classical energy of an Euler-Bernoulli beam subject to the local rotations and the stretching of the beam axis in the reference configuration:

$$W_e^{\text{corot}} = W_b \left(\frac{\|\mathbf{x}_2 - \mathbf{x}_1\|}{L} - 1, \boldsymbol{\theta}^{\text{loc}} \right) \quad (3.29)$$

with W_b given by Eq. (3.13).

The corresponding force and stiffness matrices,

$$\mathbf{F} = \mathbf{B}^T \frac{\partial W_b}{\partial \boldsymbol{\epsilon}^{\text{loc}}}, \quad \mathbf{K} = \mathbf{B}^T \frac{\partial^2 W_b}{(\partial \boldsymbol{\epsilon}^{\text{loc}})^2} \mathbf{B} + \mathbf{F} \cdot \mathbf{C}, \quad (3.30)$$

where \mathbf{B} and \mathbf{C} are the partial derivatives of the strains w.r.t. the global dofs as in Eq. (3.10).

One difference between the formulation of Crisfield [31] and ours is the updating of the rotation vectors $\boldsymbol{\theta}^{ud}$. Theirs used forces that are conjugate to the spatial spin variables $\boldsymbol{\nu}$, which correspond to an infinitesimal rotation superimposed onto the global rotation matrix, requiring a non-additive update of the form

$$\mathbf{R}(\boldsymbol{\theta}^{i+1}) = \mathbf{R}(\delta \boldsymbol{\nu}) \mathbf{R}(\boldsymbol{\theta}^i). \quad (3.31)$$

However, we follow Battini and Pacoste [11] in using additive updates of the global rotational degrees of freedom, i.e. $\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^i + \delta \boldsymbol{\theta}$ (this results in the

forces and stiffness matrices shown in Eq. (3.30)). We note that the change from multiplicative updates to additive updates did not change the equilibrium solution nor the convergence properties when applied to the numerical benchmark tests from Crisfield [31]. The forces and stiffness matrices conjugate to the additive rotation vector updates with the 3D corotational beam element are computed using finite differences.

3.3 History-Dependent Problems

The initial fully-resolved problem statement in Eq. (3.6) is used to solve elastic quasistatic problems. Here we show that one can easily include dynamics and history-dependent material behavior in the variational formulation. The resulting formulation is similar to the quasistatic problem outlined above, and can thus be approximated using similar techniques.

3.3.1 Dissipative Materials

As shown in [72, 77], introducing history dependence through viscoplasticity into the formulation of quasistatic boundary value problems can be done while maintaining the variational formulation of the problem. The process of introducing inelastic internal variables and discretizing the principle of maximum dissipation into an incremental variational formulation is not discussed here (see e.g. [77, 86, 88]) but we give examples of two material nonlinearities and how they are incorporated into the model — namely rate-independent plasticity with isotropic hardening for bars, and brittle failure of beams.

Example: Rate-Independent Plasticity with Isotropic Hardening

As a simple example of a history-dependent material, we use a one-dimensional rate-independent plasticity model with isotropic hardening based on the model in Simo and Hughes [97]. We give an overview of the model here, but we refer the reader to [97] for a more in-depth discussion.

To model rate-independent plasticity, the elastic energy density given in Eq. (3.7) can be replaced by an *incremental energy* of the form

$$\tilde{W}(\varphi^{k+1}, \boldsymbol{\xi}^{k+1}; \boldsymbol{\xi}^k) = W_e(\varphi^{k+1}, \boldsymbol{\xi}^{k+1}; \boldsymbol{\xi}^k) + W_p(\boldsymbol{\xi}^{k+1}), \quad (3.32)$$

where $\boldsymbol{\xi}$ are internal variables that parametrize the inelastic deformation, W_p represents the stored plastic energy, the subscript $k + 1$ denotes the values of the variables at the new time step, and k denotes the previous time step. For

the case of plasticity with isotropic hardening, plastic strains and accumulated plastic strain variables (denoted by the internal variables $\boldsymbol{\xi} = \{\varepsilon_p, \gamma\}$) can be used to parametrize the plastic deformation, and

$$\begin{aligned} W_e(\boldsymbol{\varphi}^{k+1}, \boldsymbol{\xi}^{k+1}; \boldsymbol{\xi}^k) &= W(\varepsilon^{k+1} - \varepsilon_p^k - \Delta\varepsilon_p) \\ W_p(\Delta\boldsymbol{\xi}) &= \begin{cases} \frac{1}{2}K(\Delta\gamma)^2 + \sigma_y + K\gamma^k|\Delta\varepsilon_p| & \text{if } \Delta\gamma \geq |\Delta\varepsilon_p| \\ \infty & \text{otherwise,} \end{cases} \end{aligned} \quad (3.33)$$

are the corresponding elastic energy, and plastic work. In Eq. (3.33), σ_y is the initial yield stress, K is a hardening parameter, ε is the axial bar strain, and we use the shorthand Δ to represent the change in a variable from state k to $k+1$.

By introducing internal variables into the problem, we now seek to identify entire states of the system, i.e. $\{\boldsymbol{\varphi}^{k+1}, \boldsymbol{\xi}^{k+1}\}$ that minimize the total incremental potential energy. However, since the potential only depends locally on the internal variables, $\boldsymbol{\xi}^{k+1}$ can be minimized at each individual element, yielding an effective energy that only depends on the displacements

$$W_t(\boldsymbol{\varphi}^{k+1}) = \inf_{\boldsymbol{\xi}^{k+1}} \{W_t(\boldsymbol{\varphi}^{k+1}, \boldsymbol{\xi}^{k+1}; \boldsymbol{\xi}^k)\}. \quad (3.34)$$

This unconstrained minimization problem Eq. (3.34) can be recast as an equivalent constrained minimization (removing the effective constraint from the definition of the plastic work), with Lagrange multipliers λ :

$$\begin{aligned} W_t(\boldsymbol{\varphi}^{k+1}) &= \inf_{\boldsymbol{\xi}^{k+1}} \sup_{\lambda \geq 0} \left\{ W_e(\varepsilon^{k+1} - \varepsilon_p^k - \Delta\varepsilon_p) + \frac{1}{2}K(\Delta\gamma)^2 \right. \\ &\quad \left. + (\sigma_y + K\gamma^k)|\Delta\varepsilon_p| + \lambda(|\Delta\varepsilon_p| - \Delta\gamma) \right\}. \end{aligned} \quad (3.35)$$

Taking $W_e(\varepsilon) = E\varepsilon^2$ to be a linear elastic material model, the resulting optimality (also called Karush-Kuhn-Tucker or KKT) conditions are

$$\begin{aligned} 0 &= K\Delta\gamma - \lambda \\ 0 &\in -E(\varepsilon^{k+1} - \varepsilon_p^k - \Delta\varepsilon_p) + (\sigma_y + K\gamma^k + \lambda) \text{sign}(\Delta\varepsilon_p) \\ |\Delta\varepsilon_p| - \Delta\gamma &\leq 0, \quad \lambda \geq 0, \quad \lambda(|\Delta\varepsilon_p| - \Delta\gamma) = 0, \end{aligned} \quad (3.36)$$

where the last condition is often referred to as complementary slackness. Using the definition of the elastic predictor stress $\sigma^{\text{trial}} = E(\varepsilon - \varepsilon_p^k)$, and rearranging

the equations, we arrive at

$$\begin{aligned}\lambda &= K\Delta\gamma \\ \sigma^{\text{trial}} &\in -E\Delta\varepsilon_p + (\sigma_y + K\gamma^k + \lambda) \text{sign}(\Delta\varepsilon_p) \\ \Delta\gamma &\geq |\Delta\varepsilon_p|, \quad \lambda \geq 0, \quad \lambda(|\Delta\varepsilon_p| - \Delta\gamma) = 0.\end{aligned}\tag{3.37}$$

The complementary slackness condition leads to two cases: The first case is when $\lambda = 0$ and $\Delta\gamma \geq |\Delta\varepsilon_p|$, and corresponds to the elastic step

$$\begin{aligned}\lambda &= 0 \\ \Delta\gamma &= \Delta\varepsilon_p = 0 \\ |\sigma^{\text{trial}}| &\leq \sigma_y + K\gamma^k\end{aligned}\tag{3.38}$$

and the second case corresponds to a step where plasticity occurs, and is when $\Delta\gamma = |\Delta\varepsilon_p|$ and $\lambda > 0$,

$$\begin{aligned}\Delta\gamma &= \frac{|\sigma^{\text{trial}}| - \sigma_y - K\gamma^k}{K - E} \\ \Delta\varepsilon_p &= \Delta\gamma \text{sign}(\sigma^{\text{trial}}).\end{aligned}\tag{3.39}$$

This is equivalent to the isotropic hardening plasticity model and corresponding update algorithm outlined in Simo and Hughes [97]. The effective energy that only depends on the displacements (not written out explicitly here for conciseness) has the corresponding stresses and tangent moduli (see Fig. 3.3) depending on whether the step is elastic (superscript e) or plastic (superscript p)

$$\begin{aligned}\sigma^e &= \sigma^{\text{trial}} & \sigma^p &= \left(1 - \frac{\Delta\gamma E}{|\sigma^{\text{trial}}|}\right) \sigma^{\text{trial}} \\ \mathbb{C}^e &= E & \mathbb{C}^p &= \frac{KE}{K + E}.\end{aligned}\tag{3.40}$$

Example: Brittle Failure

In order to model brittle truss lattices (e.g. glassy carbon lattices [12, 55]), it is desirable to model brittle failure of individual truss members. We follow Tankasala et al. [106] in using a failure criterion where the beam fails catastrophically when either the average tensile stress (ATS) over the cross-section or the maximum local tensile stress (LTS) in the cross-section reaches the yield stress of the constituent material, σ_y .

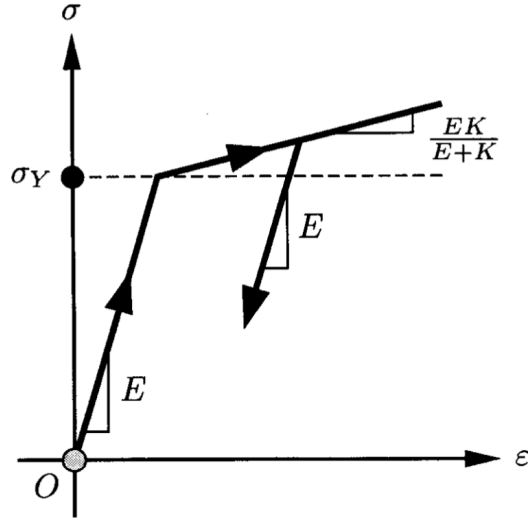


Figure 3.3: Example stress-strain curve for the plasticity model with isotropic hardening. Reprinted by permission from Springer Nature: Springer Computational Inelasticity by J.C. Simo and T.J.R. Hughes, copyright 2000 [97].

Using the local strain values defined in Section 3.2.2, the average tensile stress and maximum tensile stress in a 2D bar with thickness t can be expressed in terms of the local beam strains defined in Eq. (3.11) as

$$\sigma^{\text{ATS}} = E\varepsilon_{\text{ax}} \quad \sigma^{\text{LTS}} = E \left(\varepsilon_{\text{ax}} + \max \left(|4\tilde{\theta}_1 + 2\tilde{\theta}_2|t, |2\tilde{\theta}_1 + 4\tilde{\theta}_2|t \right) \right) \quad (3.41)$$

The expressions for the maximum stress in more complicated cross-section geometries and in 3D are more involved, and are not included here.

We use a damage parameter that can be either 0 or 1 depending on whether the beam is intact or not as the internal variable. In other words, the elastic energy of the beam is

$$\tilde{W}_b(\varepsilon^{\text{loc}}, \boldsymbol{\xi}^{k+1}) = (1 - \xi^{k+1})W_b(\varepsilon^{\text{loc}}). \quad (3.42)$$

Unlike plasticity with isotropic hardening, we do not minimize the energy with respect to the internal internal variable at each beam finite element to obtain an effective displacement problem, as this can lead to undesirable results. Instead, we perform alternating minimization, where first the equilibrium displacements are solved for assuming everything remains elastic, then one-by-one, beams are broken based on the value of their maximum stress. Algorithm 1 outlines the method used to update the damage parameter in the beams.

Algorithm 1 Update Damage Parameter

Initialize new damage parameters $\xi^{k+1} = \xi^k$
repeat
 Solve for displacements $\tilde{\varphi}^{k+1} = \arg \min_{\tilde{\varphi}} \mathcal{I}(\tilde{\varphi}; \xi^{k+1})$
 Initialize list of trusses and max stresses, $\Sigma = \{\}$
 for each truss element t **do**
 Compute maximum stress σ_t^{\max}
 if $\sigma_t^{\max} > \sigma_y$ **then**
 Add $\{\sigma_t, t\}$ to list Σ
 Get element of list with largest stress $\{\sigma_{t^*}, t^*\} = \max(\Sigma)$
 Break beam, $\xi_{t^*}^{k+1} = 1$
until Σ is empty
 Move to next load step $k = k + 1$

3.3.2 Dynamics

The corresponding dynamic problem of Eq. (3.6) is commonly written as solving the partial differential equation

$$\mathbf{M}\ddot{\tilde{\varphi}} + \frac{\partial \mathcal{I}}{\partial \tilde{\varphi}} = 0^3, \quad (3.43)$$

where \mathbf{M} represents the global mass matrix of the system, and $\partial \mathcal{I} / \partial \tilde{\varphi}$ is the nonequilibrium force. This PDE is typically solved using either an explicit or an implicit time integrating technique. One family of time-stepping techniques is the popular Newmark method, which defines the dofs and velocities at the next timestep $k + 1$ using the state at time k :

$$\begin{aligned} \tilde{\varphi}^{k+1} &= \tilde{\varphi}^k + \Delta t \dot{\tilde{\varphi}}^k + \frac{\Delta t^2}{2} ((1 - 2\beta)\ddot{\tilde{\varphi}}^k + 2\beta\ddot{\tilde{\varphi}}^{k+1}), \\ \dot{\tilde{\varphi}}^{k+1} &= \dot{\tilde{\varphi}}^k + \Delta t ((1 - \gamma)\ddot{\tilde{\varphi}}^k + \gamma\ddot{\tilde{\varphi}}^{k+1}) \end{aligned} \quad (3.44)$$

where the second derivative is defined to be

$$\ddot{\tilde{\varphi}}^k := -\mathbf{M}^{-1} \frac{\partial \mathcal{I}}{\partial \tilde{\varphi}}, \quad (3.45)$$

and β and γ are parameters chosen by the user. The method is second order accurate if and only if $\gamma = 1/2$ (and it also can be shown to follow from a discretization of Hamilton's Principle [57]), while $\gamma < 1/2$ introduces numerical damping. Common choices for β are $\beta = 1/4$ (average acceleration), $\beta = 1/6$ (constant acceleration), and $\beta = 0$ (explicit integration).

³With a slight abuse of notation, $\tilde{\varphi}$ here refers to the long vector consisting of all of the dofs of the fully-resolved truss lattice.

The update rule can be condensed to

$$\begin{aligned}\tilde{\varphi}^{k+1} &= \tilde{\varphi}^{\text{pred}} + \beta \Delta t^2 \ddot{\varphi}^{k+1} \\ \dot{\tilde{\varphi}}^{k+1} &= \dot{\tilde{\varphi}}^k + \Delta t \left((1 - \gamma) \ddot{\varphi}^k + \gamma \ddot{\varphi}^{k+1} \right),\end{aligned}\tag{3.46}$$

where $\tilde{\varphi}^{\text{pred}}$ is a predictor for the dofs at the $k + 1$ time step defined by

$$\tilde{\varphi}^{\text{pred}} = \tilde{\varphi}^k + \Delta t \dot{\tilde{\varphi}}^k + \frac{\Delta t^2}{2} \left((1 - 2\beta) \ddot{\varphi}^k \right).\tag{3.47}$$

It can be seen that the update equations for $\tilde{\varphi}^{k+1}$ are the Euler-Lagrange equations of the potential [57]

$$\mathcal{F}(\tilde{\varphi}^{k+1}) = \mathcal{I}(\tilde{\varphi}^{k+1}) + \frac{1}{2\beta\Delta t^2} (\tilde{\varphi}^{k+1} - \tilde{\varphi}^{\text{pred}})^T \mathbf{M} (\tilde{\varphi}^{k+1} - \tilde{\varphi}^{\text{pred}})\tag{3.48}$$

for $\beta \neq 0$.⁴

This means that the quasistatic problem Eq. (3.6) can be modified to include the Newmark method for dynamics by adding the incremental potential

$$\mathcal{D}(\tilde{\varphi}^{k+1}) = \frac{1}{2\beta\Delta t^2} (\tilde{\varphi}^{k+1} - \tilde{\varphi}^{\text{pred}})^T \mathbf{M} (\tilde{\varphi}^{k+1} - \tilde{\varphi}^{\text{pred}}),\tag{3.49}$$

and then updating the velocities according to Eq. (3.44), which can be computed using known quantities once the incremental displacement minimization problem is solved⁵.

The global inertial term, $\mathcal{D}(\tilde{\varphi}^{k+1})$, can be expressed as a sum over unit cells similar to the strain energy in the fully-resolved quasistatic problem with

$$\begin{aligned}\mathcal{D}(\tilde{\varphi}^{k+1}) &= \sum_{\Omega_u \in \Omega} \mathcal{D}_u, \\ \mathcal{D}_u &= \sum_{e \in \mathcal{E}_u} \mathcal{D}_e + \frac{1}{2} \sum_{e \in \mathcal{E}_u^n} \mathcal{D}_e \quad \mathcal{D}_e = (\tilde{\varphi}_e^{k+1} - \tilde{\varphi}_e^{\text{pred}})^T \mathbf{M}_e (\tilde{\varphi}_e^{k+1} - \tilde{\varphi}_e^{\text{pred}}),\end{aligned}\tag{3.50}$$

where \mathbf{M}_e is the mass matrix of the corresponding truss finite element, $\tilde{\varphi}_e \subset \{\tilde{\varphi}_u, \tilde{\varphi}_{\mathcal{N}(u)}\}$ represent the dofs of both nodes of the truss element, and \mathcal{D}_u and \mathcal{D}_e represent the inertial term of individual unit cells and truss members respectively.

⁴For the case of $\beta = 0$, \mathcal{F} can be multiplied by β to produce a well-defined potential for the dynamics problem.

⁵If the quasistatic potential is nonconvex, then in general Eq. (3.48) may not be convex and the minimization problem can be replaced by a stationarity condition.

3.3.3 Effective Displacement Problem

As shown above, history-dependent problems can also be formulated within the variational framework, and the resulting variational problem is similar to the initial quasistatic problem of Eq. (3.6). The additional history variables can either be minimized out pointwise (in the case of plasticity) or updated given the new displacements (in the case of dynamics or brittle failure), leaving an effective minimization problem that only depends on the displacements at the next time step. For the general case of dynamic problems with internal variables, the fully-resolved problem changes to the incremental problem,

$$\begin{aligned}\tilde{\varphi} &= \arg \min_{\tilde{\varphi} \in V} \tilde{\mathcal{I}}(\tilde{\varphi}) = \arg \min_{\tilde{\varphi} \in V} \left\{ \sum_{\Omega_u \in \Omega} \tilde{W}_u(\tilde{\varphi}_u, \tilde{\varphi}_{N(u)}) - \mathcal{L}(\tilde{\varphi}) \right\} \\ \tilde{W}_u(\tilde{\varphi}_u, \tilde{\varphi}_{N(u)}) &= W_u(\tilde{\varphi}_u, \tilde{\varphi}_{N(u)}) + \mathcal{D}_u(\tilde{\varphi}_u, \tilde{\varphi}_{N(u)}), \\ W_u(\tilde{\varphi}_u, \tilde{\varphi}_{N(u)}) &= \sum_{t \in \mathcal{E}_u} W_t(\varphi) + \frac{1}{2} \sum_{t \in \mathcal{E}_u^n} W_t(\varphi),\end{aligned}\tag{3.51}$$

where the implied superscript of $k + 1$ on the state variables is excluded for conciseness, and dependence on the old state, $\{\varphi^k, \dot{\varphi}^k, \ddot{\varphi}^k, \xi^k\}$ is implicitly assumed, $\tilde{\mathcal{I}}(\tilde{\varphi})$ represents the total (possibly modified with dynamics or internal variables) potential of the system, \tilde{W}_u is the (possibly modified) potential of each unit cell, \mathcal{D}_u represents the inertial component of the unit cell potential (and is given by Eq. (3.50)), and W_u is the quasistatic energy of the unit cell. For elasto-plastic trusses,

$$W_t(\varphi) = \inf_{\xi} \{W_t(\varphi, \xi)\}\tag{3.52}$$

is the effective truss potential, such that the entire problem can be stated in terms of only the new displacement field. For trusses undergoing brittle failure,

$$W_t(\varphi) = W_t(\varphi; \xi)\tag{3.53}$$

is the "elastic predictor" energy of the lattice (i.e. the energy computed with the internal variables from the previous step), and the internal variables are updated one at a time according to Algorithm 1.

Whether modeling quasistatic or history-dependent trusses, we will use \mathcal{I} to denote the total potential of the system, W_u to be the unit cell potential (including the inertial term if performing dynamic analysis) and W_t to be the element energy.

3.4 The Quasicontinuum Method

The fully resolved lattice problem Eq. (3.6) is computationally expensive for two reasons:

1. The number of dofs in the system can be extremely large, requiring an extremely expensive solve (e.g. a large linear system solve if using Newton's method)
2. The computation of the energy (or equivalently, the force or stiffness matrix) requires a sum over all unit cells in the truss.

The QC method systematically introduces kinematic constraints and energy approximation techniques to reduce the computational costs associated with Item 1 and Item 2.

3.4.1 Kinematic Constraints

In the spirit of the original QC method, we approximate the deformation of a truss lattice by a reduced set of dofs. However, instead of reducing the dofs by selecting representative nodes (analogous to the introduction of representative atoms in the original QC method), we choose a set of $N_{\text{rep}} \ll N$ *representative unit cells* (in the following called *repUCs*, not to be confused with representative volume elements or unit cells commonly used in homogenization), which retain all of their dofs.

While the dofs of the repUCs are retained, the dofs of the remaining UCs are obtained from interpolation as in the original QC method:

$$\tilde{\varphi}(\mathbf{X}_u) = \sum_{r=1}^{N_{\text{rep}}} N_r(\mathbf{X}_u) \tilde{\varphi}_r \quad (3.54)$$

with some suitable shape functions $N_r(\mathbf{X})$, and the corresponding dofs of the repUCs, $\tilde{\varphi}_r$. In the following, we mainly focus on using linear interpolation on a Delaunay-triangulated mesh. The simplices of the Delaunay-triangulated mesh will be referred to as *coarse-grained* or *macroscopic* elements in order to differentiate them from the beam or bar elements used to model the individual truss members within the unit cells. Note that the shape functions are evaluated at the location \mathbf{X}_u of unit cell u (in the reference configuration), and not at the exact positions of each individual node inside the unit cell; see Fig. 3.4. This means that some unit cells that lie inside — but near the edge

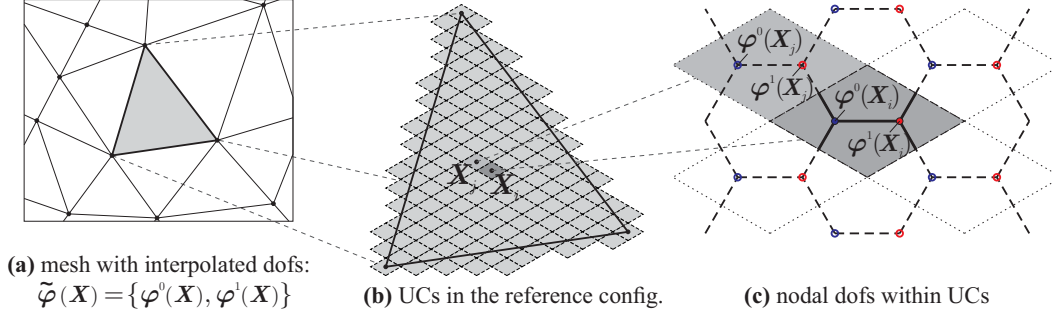


Figure 3.4: Illustration of the interpolated dofs and the recovery of nodal dofs within individual unit cells from the interpolation: (a) shows a mesh of coarse-grained/macroscopic elements whose vertices are repUCs that carry dofs $\tilde{\varphi}$. (b) Within an element K , the dofs $\tilde{\varphi}$ are interpolated by evaluating the shape functions at each unit cell location, e.g. \mathbf{X}_i or \mathbf{X}_j (in the reference configuration). (c) Dofs $\{\varphi^0, \varphi^1\}$ of the two nodes within each unit cell are recovered from the interpolated repUC dofs.

— of some macroscopic element K can contain some nodes that lie outside of K , but still use the shape functions of K to interpolate the dofs (since the node is contained in a unit cell that lies within K).

An equivalent way of interpreting the shape functions is by thinking of each Bravais lattice in the multilattice being interpolated with the same shape functions, but shifted relative to one another by the corresponding shift vector of the lattice.

By imposing the kinematic constraints in Eq. (3.54), the full-resolved problem is simplified to

$$\inf_{\tilde{\varphi} \in V^h} \mathcal{I}(\tilde{\varphi}), \quad (3.55)$$

where V^h now represents the (much smaller) set of dofs defined by the shape functions N_r that satisfy the essential boundary conditions.

3.4.2 Energy Approximation

The above quasicontinuum approximation reduces the number of dofs in the system, whereas the efficient calculation of the total energy \mathcal{I} necessitates the introduction of sampling rules (also referred to as summation rules) similar to quadrature in finite elements. To this end, we select $N_s \ll N$ sampling unit cells such that

$$\mathcal{I}(\tilde{\varphi}) \approx \mathcal{I}^h(\tilde{\varphi}) = \sum_{s=1}^{N_s} w_s W_s(\tilde{\varphi}) - \mathcal{L}_s(\tilde{\varphi}) \quad (3.56)$$

with weights w_s akin to quadrature weights, and \mathcal{L}_s is the sampled external force potential. Various summations rules have been introduced for atomistic and truss-based QC – see e.g. [3, 17, 37, 49, 58]. In the following, we use variants of the first-order and second-order optimal summation rules proposed in Amelang et al. [3] modified by features from the central summation rule of Beex et al. [17].

Before outlining the summation rules used in this thesis, we first take the time here to go over some concepts that will be used in the formulation. The first key concept is to notice that in the Bravais coordinates (see Eq. (3.2)), all of the repUC's have integer coordinates, and thus, lie on the integer lattice. The mapping from the cartesian to Bravais coordinates is given by Eq. (3.3), and has the jacobian $J^{-1} = \det(\mathbf{A}) = \text{vol}(\Omega_{\text{UC}})$ such that

$$\text{vol}(K_{\text{int}}) = \frac{\text{vol}(K)}{\text{vol}(\Omega_{\text{UC}})}, \quad (3.57)$$

where $\text{vol}(K_{\text{int}})$ is the "volume" of the simplex K when expressed in Bravais coordinates.

The other important concept is that of solid angles. We define the solid angle $\omega_K(\mathbf{Y})$ of a point \mathbf{Y} with respect to element K as the fraction of a ball with infinitesimal radius centered at \mathbf{Y} that is inside K , i.e.

$$\omega_K(\mathbf{Y}) = \lim_{r \rightarrow 0} \frac{\text{vol}(B(r, \mathbf{Y}) \cap K)}{\text{vol}(B(r, \mathbf{Y}))}, \quad (3.58)$$

where $\text{vol}(B(r, \mathbf{Y}))$ corresponds to the volume of the d -dimensional ball B with radius r centered at \mathbf{Y} . For example, the solid angle for a point with respect to a triangle is

$$\omega_K(\mathbf{Y}) = \begin{cases} 1 & \text{if } \mathbf{Y} \text{ is in the interior of } K \\ 1/2 & \text{if } \mathbf{Y} \text{ is on the edge of } K \\ \theta/(2\pi) & \text{if } \mathbf{Y} \text{ is a vertex of } K \\ 0 & \text{if } \mathbf{Y} \text{ is outside of } K \end{cases}, \quad (3.59)$$

where θ is the angle at the vertex. The solid angle of points are useful because the solid angle weighted sum over all of the integer points of an element with integer coordinates K_{int} (i.e. all of the simplices of the mesh in Bravais coordinates) can be used to approximate the volume of the element (in Bravais coordinates)

$$\text{vol}(K_{\text{int}}) \approx \sum_{\mathbf{Y} \in K_{\text{int}} \cap \mathbb{Z}^d} \omega_{K_{\text{int}}}(\mathbf{Y}). \quad (3.60)$$

In two dimensions, the approximation Eq. (3.60) for general integer polytopes becomes an equality, and can be expressed as [84]

$$\text{vol}(\mathcal{P}_{\text{int}}) = \#i + \frac{\#b}{2} - 1, \quad (3.61)$$

where $\#i$ represents the number of lattice points on the interior of the integer polytope \mathcal{P}_{int} , and $\#b$ is the number of lattice points on the boundary of \mathcal{P}_{int} . This is commonly known as Pick's theorem. Applying Pick's theorem to a triangle, one obtains

$$\text{vol}(K_{\text{int}}) = \#i + \frac{\#e}{2} + 1/2, \quad (3.62)$$

where $\#e$ is the number of non-vertex lattice points on the boundary (i.e. lattice points on the edges of the triangle).

3.4.3 First-Order Summation Rule

The sampling unit cells in the first-order summation rule used here are chosen such that all repUCs are also sampling unit cells with a weight of $w_{s,\text{rep}} = 1$ (as in [17]). In other words, each repUC's energy is computed exactly, and it is not used to approximate the energy of other unit cells. In order to approximate the energy of all of the other unit cells, an additional sampling site is placed at the barycenter of each macroscopic simplicial element, K . Unlike in Beex et al. [17], the barycenter sampling unit cell here computes the dofs of its neighbors using the first-order expansion

$$\tilde{\varphi}_{\text{loc}}(\mathbf{X}_i) = \tilde{\varphi}(\mathbf{X}_s) + \left. \frac{\tilde{\varphi}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}_s} (\mathbf{X}_i - \mathbf{X}_s), \quad (3.63)$$

where \mathbf{X}_s is the location of the sampling unit cell, and \mathbf{X}_i is the location of the neighboring unit cell. We note that if the lattice is a Bravais lattice, this local approximation is equivalent to the commonly-used Cauchy-Born rule, as in [3]. For multilattices, the Cauchy-Born rule usually consists of solving a microscale problem to compute the energy, which differs from our formulation. Therefore, we refer to these unit cells simply as *local* sampling unit cells. The local sampling unit cell attempts to approximate the energy of the remaining non-repUCs in the macroscopic element, and therefore its weight is chosen to approximate the number of non-repUCs in the element K :

$$w_{s,\text{inner}} = \frac{\text{vol}(K)}{\text{vol}(\Omega_{\text{UC}})} - \sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}), \quad (3.64)$$

where $\text{vol}(K)$ and $\text{vol}(\Omega_{\text{UC}})$ are the volumes of the macroscopic element in cartesian coordinates and of the unit cell, respectively, and $\omega_K(\mathbf{X}_i^{\text{rep}})$ is the solid angle of the i^{th} repUC. It should be noted here that since $\sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}) = 1/2$ in 2D and the fact that $\text{vol}(K)/\text{vol}(\Omega_{\text{UC}})$ is the volume of the integer simplex in Bravias coordinates $\text{vol}(K_{\text{int}})$, the barycenter sampling weight naturally goes to zero for 2D meshes in fully-resolved regions (i.e. where there are no interior or non-vertex boundary unit cells).

However, since Pick's theorem does not hold in higher dimensions⁶, the sampling weight computed by Eq. (3.64) does not naturally go to zero in fully-resolved regions in 3D. Therefore, barycenter sampling weights that fall below a chosen threshold are manually set to zero. This results in barycenter sampling unit cells only being used in non-fully-resolved regions. Figure 3.5 shows an example lattice with sampling unit cells of the first-order summation rule described here.

3.4.4 Second-Order Summation Rule

The second-order summation rule expands on the first-order summation rule by adding additional sampling unit cells at the centroids of the edges or faces of the simplices in 2D or 3D, respectively, as was done in [3]. These sampling unit cells are given weights equal to the number of unit cells that lie exactly on the edge (or face in 3D) excluding the boundaries.

Counting the number of unit cells on edges of triangle and faces of tetrahedra can be done in a computational efficient manner (i.e. without enumerating all of the unit cells), such that computing the sample unit cell weights does not become impractical for very coarse meshes. The details of how these weights are computed has been moved to Appendix B.

Since the second-order rule introduces more sampling unit cells, the weights of the barycenter sampling unit cells are adjusted accordingly:

$$w_{s,\text{inner}} = \frac{\text{vol}(K)}{\text{vol}(\Omega_{\text{UC}})} - \sum_{i=1} w_{i,\text{edge}} \omega_K(\mathbf{X}_i^{\text{edge}}) - \sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}), \quad (3.65)$$

where $\omega_K(\mathbf{X}_i^{\text{edge}})$ is the solid angle of the i^{th} edge (or face in 3D) sampling unit cell. It should be noted here that in 2D, since the edge weights exactly

⁶Reeve [84] showed that a higher-dimensional versions of Pick's theorem does not exist by demonstrating that a lattice tetrahedron can have an arbitrarily large volume without containing any non-vertex lattice points. The example tetrahedron is commonly called Reeve's tetrahedron.

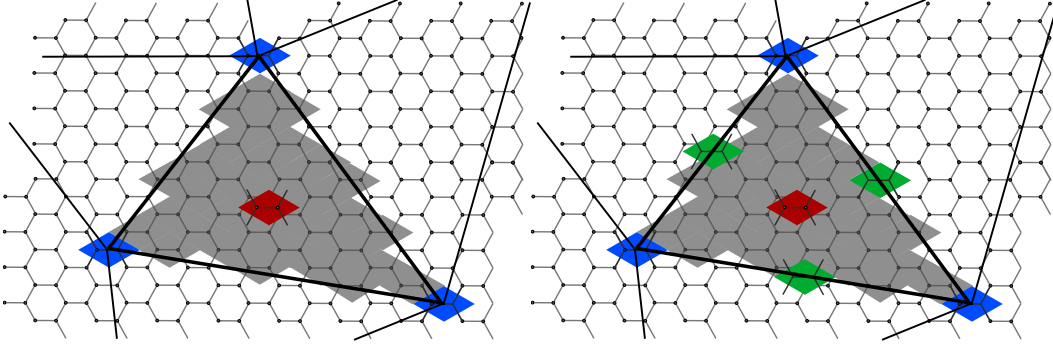


Figure 3.5: First and second-order summation rules. The first-order summation rule (left) shows the vertex sampling unit cells (blue) and the local inner sampling unit cell (red). The second-order summation rule (right) also has edge sampling unit cells (green).

count the number of unit cells on the edges, Pick's theorem and Eq. (3.60) can be used to show that the barycenter sampling unit cell is exactly equal to the number of unit cells on the interior of K . From Eq. (3.65) and (3.62), we arrive at

$$\begin{aligned} w_{s,\text{inner}} &= \left(\#i + \frac{\#e}{2} + \frac{1}{2} \right) - \left(\frac{\#e}{2} \right) - \left(\frac{1}{2} \right) \\ &= \#i. \end{aligned} \quad (3.66)$$

3.4.5 Consistency of the Summation Rules

In order for a summation rule to be 0^{th} -order consistent, the sum of the sample unit cell weights must equal the number of unit cells in the simulation. For the first-order summation rule in two dimensions, the sum of the sampling weights is

$$\begin{aligned} \sum_s w_s &= \sum_K w_{s,\text{inner}} + \sum_{\text{rep}} w_{s,\text{rep}} \\ &= \sum_K \left(\frac{\text{vol}(K)}{\text{vol}(\Omega_{\text{UC}})} - \sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}) \right) + \sum_{\Omega_{\text{rep}}} w_{s,\text{rep}} \\ &= \frac{\text{vol}(\Omega)}{\text{vol}(\Omega_{\text{UC}})} - \sum_K \sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}) + \sum_{\Omega_{\text{rep}}} w_{s,\text{rep}} \\ &= \#i + \frac{\#b}{2} - 1 + \sum_{\partial\Omega \cap \Omega_{\text{rep}}} 1 - \omega_K(\mathbf{X}^{\text{rep}}), \end{aligned} \quad (3.67)$$

where Ω_{rep} is the set of repUCs, and $\partial\Omega \cap \Omega_{\text{rep}}$ is the set of repUCs that lie on the boundary of the geometry, so that

$$\sum_s w_s - (\#i + \#b) = - \left(\left(\frac{\#b}{2} - 1 \right) - \sum_{\partial\Omega \cap \Omega_{\text{rep}}} 1 - \omega_K(\mathbf{X}^{\text{rep}}) \right). \quad (3.68)$$

In other words, the sum of sample unit cell weights differs from the total number of unit cells by the amount of the right hand side of the equation. For coarse meshes, The number of unit cells on the boundary, $\#b$ far outweighs the number of repUCs on the boundary, so the difference between the sum of the sample weights and the total number of unit cells is roughly

$$\sum_s w_s - (\#i + \#b) \approx -\frac{\#b}{2}. \quad (3.69)$$

For large geometries where the number of boundary unit cells is small compared to the number of total unit cells, the relative error approach zero, and therefore the first-order summation rule is almost 0th-order consistent.

On the other hand, the sum of the second-order sampling weights in 2D is exactly 0th-order consistent. As shown in Eq. (3.66), the barycenter weights sum up to the number of unit cells on the interior of the coarse-grained elements, the edge sampling unit cell weights sum up to the total number of unit cells that lie on coarse-grained element edges, and the repUC sampling unit cells account for the unit cells that lie on coarse-grained element vertices.

In three dimensions, neither the first-order nor the second-order summation rule is 0th-order consistent, but similar arguments to those made for the 2D summation rule can be used to show that the second-order summation rule does a better job at approximating the number of unit cells in the simulation, especially for geometries that have a significant number of unit cells on the boundary compared to the number of interior unit cells.

It is worth noting that other popular summation rules in the QC method such as cluster and quadrature summation rules are also not 0th-order consistent. These methods choose the sampling weights such that the sum of the sampling weights equals the volume of the domain (e.g. through volume-weighted quadrature rules or voronoi tessellation) normalized by the volume of the unit cell. As seen above, this causes the summation rule to underpredict the number of unit cells in the simulations (or equivalently, the energy of an affinely

strained system), but this error becomes negligible for large domains where the number of interior unit cells far outweighs the number of boundary unit cells.

3.4.6 Force Artifacts

All of the summation rules used in the QC literature produce error when approximating the total energy of the system. One way that this error can be seen is through so-called *force artifacts*, i.e. forces that occur only because of the energy approximation error. Force artifacts are most obvious when occurring in the undeformed configuration in atomic lattices. Intuitively, they represent the fact that the energy of the system can artificially be decreased by taking advantage of the inexact energy approximation. The force on a given lattice site u in the QC method can be written as

$$\mathbf{F}_u^h = \frac{\partial \mathcal{I}^h}{\partial \tilde{\boldsymbol{\varphi}}_u} = \sum_s w_s \frac{\partial W_u(\tilde{\boldsymbol{\varphi}}_s, \tilde{\boldsymbol{\varphi}}_{N(s)})}{\partial \tilde{\boldsymbol{\varphi}}_u} = \sum_s w_s \mathbf{F}_s, \quad (3.70)$$

where \mathbf{F}_s represents the force contribution from sample site s . In atomic lattices, $\mathbf{F}_s \neq \mathbf{0}$ in the undeformed state, and therefore, it is possible that $\mathbf{F}_u^h \neq \mathbf{0}$ in perfect lattices, resulting in force artifacts called *ghost forces*. As shown in Amelang et al. [3] the optimal summation rules outlined here exhibited the smallest force artifacts when compared with other popular summation rules in atomistic QC.

In the case of truss lattices, $\mathbf{F}_s = \mathbf{0}$ in the undeformed configuration, so ghost forces do not appear. However, in trusses undergoing affine deformation, force artifacts do appear analogous to ghost forces in atomistic QC.

It is common to look at the force artifacts in the QC method to gauge the error in the summation rule, but this will not be done here. Solely looking at force artifacts in trusses could be misleading. For example, a large force artifact in a stretching-dominated truss lattice would not be as severe as a large force artifact in a bending-dominated truss lattice due to the fact that bending-dominated lattices are much softer. Furthermore, in lattices like the square lattice that has bending- and stretching-dominated deformation modes, the direction of the force artifacts would also affect the error in the model (i.e. force artifacts acting in the direction of bending-dominated deformation modes will cause larger errors than artifacts acting in the direction of stretching-

dominated modes). For this reason, we choose to test the QC methodology using (among others) the elastic constants test in Section 4.1.

3.5 Adaptive Refinement

A key benefit of the fully-nonlocal QC method over other concurrent scale-bridging techniques is that no conceptual distinction is made between the fully-resolved and coarse-grained regions, allowing for automatic adaptive refinement of the discretization down to the fully-resolved limit. Note that we here constrain ourselves to adaptive refinement without coarsening (see the discussion in Tembhekar et al. [108]). Adaptivity thus requires a refinement criterion and a geometric refinement algorithm. We generally identify a coarse-grained element K for refinement if it passes the refinement criterion

$$F(K)f(\mathbf{F}_K) > r_0, \quad (3.71)$$

where $F(K)$ is some weighting function depending on the geometry of K and $f(\mathbf{F}_K)$ is a metric depending on the deformation gradient inside the element (averaged over all Bravais lattices in the multilattice, each of which is constant by the affine interpolation), and r_0 defines the refinement threshold. The specific weighting function and refinement metric are chosen based on the lattice architecture and the problem to be simulated.

Once all coarse-grained elements have been identified for refinement, we apply a longest-edge bisection algorithm [108] to insert new repUCs with the complicating constraint that all repUCs are to be located on valid lattice sites. To this end, we insert a new repUC at the vacant unit cell location nearest to the mid-point of the longest edge of each element to be refined. Once all new repUCs have been created, we recreate the element connectivity. The nodal dofs of newly inserted repUCs are computed by interpolation from the previously existing mesh (this includes the previous velocities and accelerations if dynamics is being modeled).

If plasticity or failure is included in the model, then the internal variables from the previous mesh must be transferred to the updated mesh. In order to avoid any complications with interpolating internal variables at sample unit cell locations, the simulations either do not include refinement or use a refinement criterion such that the mesh becomes fully-resolved in any regions where material nonlinear occurs. In the latter case, the internal variables are only

copied to the sample unit cells in the new mesh in the fully-resolved region, and no interpolation takes place.

3.6 Comparison with FE² and classical homogenization

The truss QC theory described above was specifically designed to accurately capture the response of truss lattices both in the fully-resolved limit and in the limit of large coarse-grained elements. In the fully-resolved limit, all dofs of all unit cells are retained and the energy of each truss member is calculated exactly, so that the lattice is described exactly⁷. However, when coarse-grained elements are large so that a separation of scales applies between the macroscopic BVP and the underlying unit cell, it is less obvious that our QC method accurately captures the truss response.

The decision to keep all of the dofs of all of the nodes inside of the unit cell as dofs at the macroscale was inspired by homogenization. As shown in the two-scale elasticity problem in Section 1.4, the effective response of the macroscopic material is achieved when allowing the microstructure to "relax". By having all of the dofs of the unit cell at each repUC site, the nodes in the unit cell are able to shift relative to one another, allowing the microstructure to reach a lower energy. The hope is that allowing the nodes in each unit cell to shift, we will be able to match the relaxed energy that is predicted by homogenization in coarse-grained regions. Of course, at this point the claim that keeping all of the unit cell dofs at the macroscale allows the method to behave similarly to homogenization when there is a separation of scales is unfounded, which is why numerous tests are performed in Chapter 4 to determine the accuracy of the method. However, it is clear that if representative nodes were used instead of repUCs, then the affine interpolation imposed on the underlying lattice could over-constrain bending-dominated multilattices such that they appear to be stretching-dominated. For example, the bending-dominated hexagon unit cell cannot undergo affine deformation (i.e. without allowing the microstructure to relax) without stretching of the beams.

Even though the proposed QC methodology does not come with the guarantees of homogenization theory when there is a separation of scales, unlike the

⁷By *exactly* we refer to exact agreement with a discrete simulation fully resolving each individual truss member. The particular choice of the underlying discrete truss description is not the topic here. It is assumed that the corotational beam formulation used to represent the truss lattice members is adequate (it can easily be replaced by alternative constitutive or kinematic descriptions on the individual beam level).

hierarchical multiscale method, the displacements here are conforming. The conformity of the displacements means that the computed solution lies in the solution space of the fully-resolved problem, and that we retain an upper bound on the energy of the original problem (i.e. up to the energy approximation by summation rules).

Chapter 4

APPLICATIONS

A significant portion of this chapter is taken from our recent publication:

Gregory P. Phlipot and Dennis M. Kochmann. A quasicontinuum theory for the nonlinear mechanical response of general periodic truss lattices. *Journal of the Mechanics and Physics of Solids*, 124:758 – 780, 2019. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2018.11.014>. URL <http://www.sciencedirect.com/science/article/pii/S0022509618304071>.

4.1 Elastic Constants

Since the linear elastic moduli of a periodic lattice generally emerge from affine states of deformation (depending on lattice architecture possibly with shifts in between affinely strained multiple Bravais lattices) [112], the multi-lattice truss QC formulation introduced above is expected to reproduce the effective, homogenized elastic moduli of, in principle, arbitrary lattice topologies. However, as was already observed in atomic lattices [3], the unstructured coarse-graining of the QC methodology introduces errors that depend on the mesh quality and the choice of the summation rule to approximate the strain energy, cf. Eq. (3.56). In particular, meshes that undergo adaptive refinement involve a wide spread in element size and quality. Therefore, it is essential to verify the accuracy of the above truss QC scheme before preceding to more advanced boundary value problems. As a worst-case scenario, we consider random meshes generated by randomly selecting repUCs from the unit cells within a lattice filling the macroscopic shape of a cube. Of course, any sensible refinement algorithm, including the one outlined in Section 3.5, will not result in random meshes but in gradually refined meshes of superior quality. The generation of random meshes is therefore viewed as a worst-case scenario in terms of mesh quality. If a method performs well here, it is expected to perform even better on adaptively refined meshes. If a method fails here, then it cannot be expected to perform well with adaptive refinement or in general unstructured meshes.

In order to understand how the method performs at various levels of coarsening, we compute the elastic moduli of a variety of 2D and 3D lattices, using

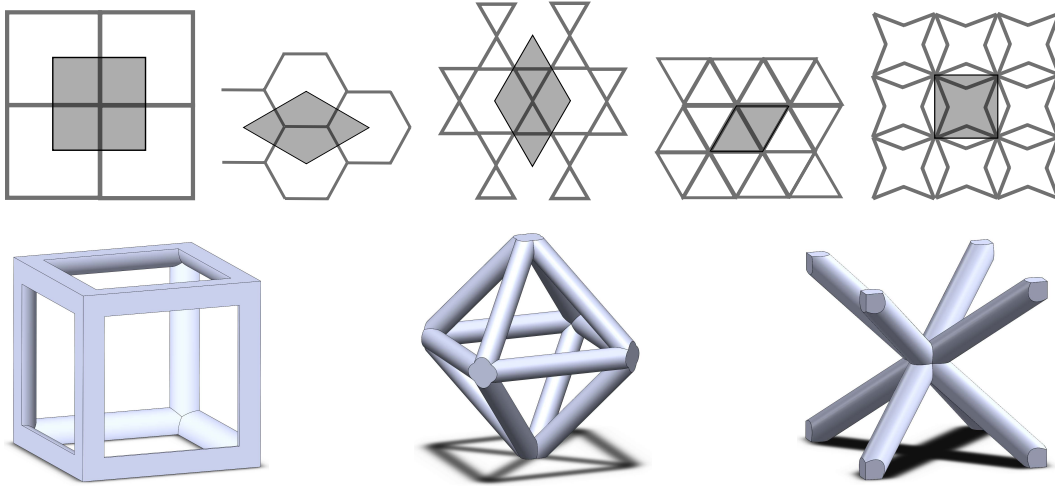


Figure 4.1: Unit cells of lattices with (a) square, (b) hexagon, (c) kagome, (d) triangle, (e) star-shaped 2D, (f) cubic, (g) cuboctahedron, and (h) BCC architecture.

meshes of various repUC densities (where by *repUC density* we denote the ratio of repUCs to total unit cells in the simulation). Specifically, we present the elastic moduli of the following lattices: square, hexagon, triangle, kagome, and star-shaped lattices in 2D, and cubic, cuboctahedron, and a bcc lattice with nearest-neighbor connectivity in 3D (see Fig. 4.1). Since the purpose of this simulation is to investigate the method in the linear elastic regime, standard Euler-Bernoulli beam elements with linearized kinematics are used to model the individual truss members.

For each lattice, the elastic moduli are computed by first affinely applying pure-shear, volumetric, and uniaxial states of strain to the translational dofs of the periphery of a finite-size lattice. To avoid complications because of unknown boundary conditions to be applied to the rotational dofs for given states of affine straining, we evaluate the strain energy for the elastic modulus extraction only in an inner block of the simulation domain (see Fig. 4.2). This way, potential boundary effects are ensured to have decayed through a peripheral region towards the center of the block.

The affine displacements applied to the translational dofs of the repUCs in the

periphery for the three load cases are given by

$$\mathbf{x}^{shear} = \begin{bmatrix} 1 & \varepsilon & 0 \\ \varepsilon & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}, \quad \mathbf{x}^{vol} = \begin{bmatrix} 1 + \frac{\varepsilon}{3} & 0 & 0 \\ 0 & 1 + \frac{\varepsilon}{3} & 0 \\ 0 & 0 & 1 + \frac{\varepsilon}{3} \end{bmatrix} \mathbf{X}, \quad \mathbf{x}^{uni} = \begin{bmatrix} 1 + \varepsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X} \quad (4.1)$$

in 3D, and

$$\mathbf{x}^{shear} = \begin{bmatrix} 1 & \varepsilon \\ \varepsilon & 1 \end{bmatrix} \mathbf{X}, \quad \mathbf{x}^{vol} = \begin{bmatrix} 1 + \frac{\varepsilon}{2} & 0 \\ 0 & 1 + \frac{\varepsilon}{2} \end{bmatrix} \mathbf{X}, \quad \mathbf{x}^{uni} = \begin{bmatrix} 1 + \varepsilon & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X} \quad (4.2)$$

in 2D. The cube is mechanically equilibrated, after which the average strain energy of the inner block of the simulation domain (see Fig. 4.2) is computed as a weighted sum over the sampling unit cells according to the first- and second-order sampling rules introduced in Section 3.4.2. Since the simulation is linear elastic, the energy is quadratic in the macroscopic strains, and the relation between the elastic modulus and the energy is

$$\lambda = \frac{2E_b}{V_b \varepsilon^2}, \quad (4.3)$$

where λ represents the shear, volumetric, or uniaxial strain elastic modulus, respectively, for the three above load cases. E_b is the total energy of the sampling sites inside the inner block, $V_b = w_s \text{vol}(\Omega_{UC})$ is the volume represented by the sampling sites with $\text{vol}(\Omega_{UC})$ being the volume of a single unit cell, and ε is the applied strain in Eq. (4.1) and Eq. (4.2).

In order to understand how the method performs at various levels of coarsening, the elastic moduli of each lattice were computed on meshes of various repUC densities from 1-100%. In addition, each simulation was repeated for two different beam slenderness ratios expressed through the relative density, viz., $\bar{\rho} = 1\%$ (slender beams) and $\bar{\rho} = 20\%$ (thicker beams). Each combination of lattice topology, relative density, and repUC density was performed on 20 different randomly generated meshes, where the overall size of each simulation was adjusted to keep a similar number of repUCs in all simulations.

The errors in the elastic moduli computed using truss QC (compared to the elastic moduli obtained from fully-resolved simulations) are summarized in Fig. 4.3 and Fig. 4.4. Overall, our truss QC method with optimal summation rules does an excellent job at predicting the elastic moduli in the fully-resolved limit (repUC density of 100%) and in the large-element limit (vanishing repUC

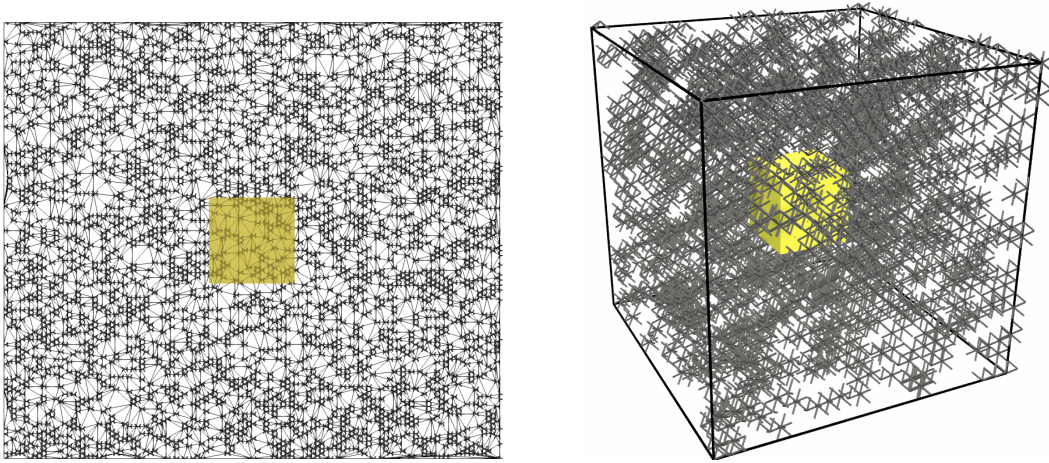


Figure 4.2: Representative, randomly coarse-grained truss QC simulations in 2D and 3D for the elastic constant calculation: (*left*) a coarse-grained hexagon lattice with 30% repUC density and (*right*) a bcc lattice with 10% repUC density. Affine boundary conditions are applied to the translational dofs on the outer faces of the block, and the energy of all sample unit cells inside the yellow box is used to calculate the elastic modulus (the large region in between removes any boundary layer effects).

density). Simulations with intermediate repUC densities show errors, with apparent variations between different lattice topologies and relative densities. Largest errors (all less than 6%) appear above repUC densities above 60%, which amounts to regions being close to fully refined and which also agrees with observations found on random atomistic QC discretizations [3]. In the regions near full resolution, the second-order summation rule shows less error than the first-order rule, but there is a tradeoff with computational expense since it requires the use of more sampling unit cells.

Indeed, it can be expected that simulations with a slightly coarsened mesh and an approximate summation rule do not perfectly match the fully resolved data, as the energy-based QC summation rules are known to generate force artifacts that lead to errors. However, the same simulations performed on uniformly coarsened meshes produce the exact elastic constants for all configurations tested, again analogous to atomistic QC [3]. This confirms that the errors in Fig. 4.3 and Fig. 4.4 are expected to be larger than those occurring in most simulations with more structured, better conditioned meshes.

Finally, in order to isolate the error source, the above simulations were performed with an exact energy summation (i.e. unit cell positions are still being

2D Lattice Moduli Errors

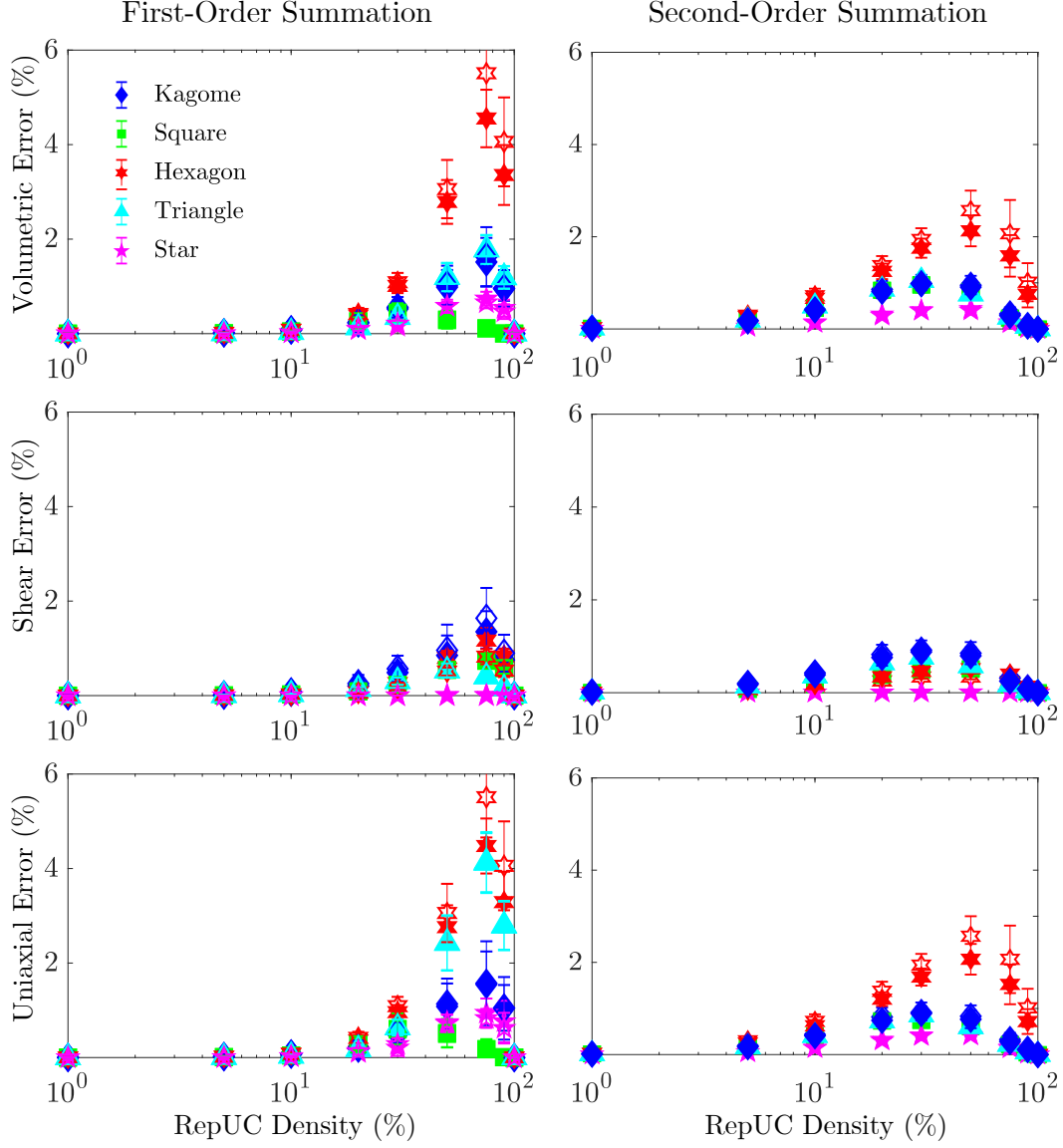


Figure 4.3: Errors in the linear elastic bulk, shear, and uniaxial moduli of 2D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with the first-order (*left*) and second-order (*right*) optimal summation rule and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.

3D Lattice Moduli Errors

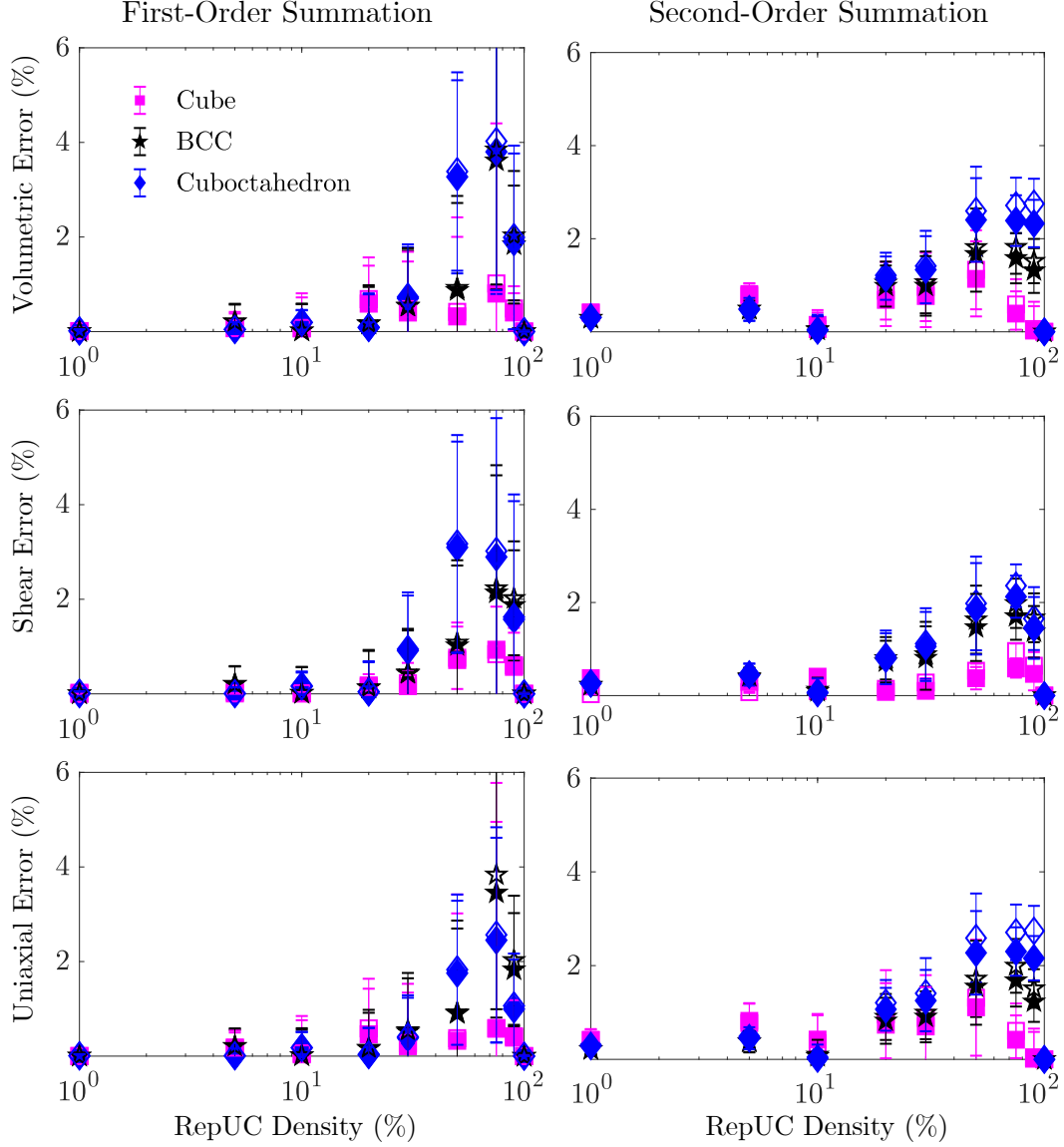


Figure 4.4: Errors in the linear elastic bulk, shear, and uniaxial moduli of 3D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with the first-order (*left*) and second-order (*right*) optimal summation rule and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.

interpolated but the total energy is calculated exactly from all unit cells in the simulation). Across all tested relative densities, lattice topologies, and repUC densities, this yields the exact elastic constants (even with randomly placed unit cells), which shows that the errors reported can be solely attributed to the approximation of the energy, and not the QC interpolation scheme.

4.2 Boundary Layer Method

Calculating the fracture toughness of truss lattices is a prime example to demonstrate the efficiency and accuracy of the truss QC method. Unlike prior techniques, truss QC allows us to restrict full resolution to those regions of interest near the crack tip while efficiently coarse-graining the remaining domain, thus easily admitting the application of remote boundary conditions.

We base our investigation on a coarse-grained version of boundary layer analysis proposed by Schmidt and Fleck [94] to compute the brittle fracture toughness of multiple lattice topologies, and we compare the results to those obtained by brute-force discrete lattice calculation. Boundary layer analysis consists of applying displacements corresponding to the K -field of an equivalent continuum to the periphery of a large, notched lattice. Since the lattices are not necessarily isotropic, the expressions for the K -Field displacements of anisotropic media from Sih et al. [96] are used (not written out here for conciseness). Brittle fracture is assumed to occur when the maximum tensile stress of any beam near the notch tip reaches the tensile strength of the constituent material within the linear elastic regime (so that linear Euler-Bernoulli beams with linearized kinematics can be used to model individual truss members).

For comparison, Schmidt and Fleck [94] conducted boundary layer analyses on circular meshes with a radius of about $85l$; Quintana Alonso and Fleck [82] and Romijn and Fleck [89] used square meshes with dimensions $600l \times 600l$ and Tankasala et al. [106] used square meshes with dimensions $500l \times 500l$, where l is the length of a beam in the lattice (all beams in the lattices have the same length). Here, we start with an initial fully-resolved region of $16l \times 16l$, surrounded by 6 regions of 6 cells, each region coarsened by a factor of two compared to the previous region for total grid size of $768l \times 768l$ (see Fig. 4.5).

We use the adaptive refinement described in Section 3.5 with the weighting function

$$F(K) = \text{vol}(K)^{1/d}, \quad (4.4)$$

where $d = 2$ is the spatial dimension, and the refinement metric based on the second invariant of the average deformation gradient of the coarse-grained elements,

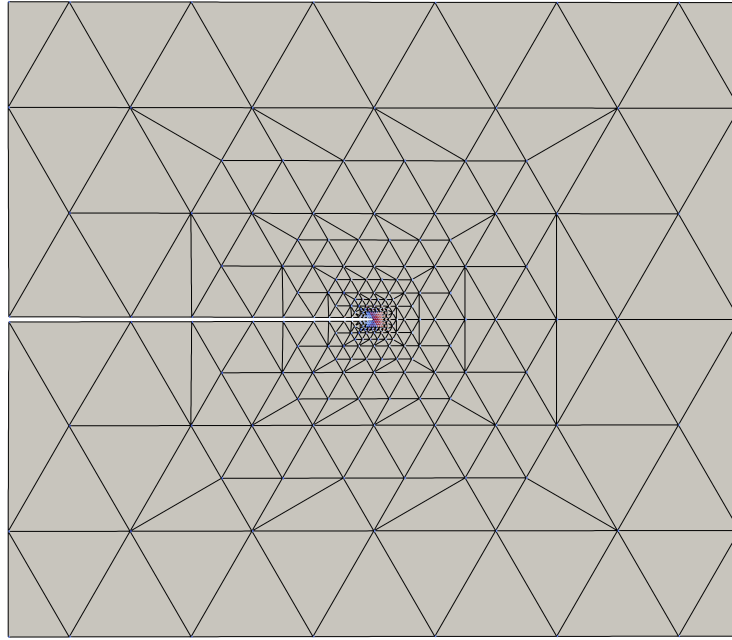
$$f(\mathbf{F}) = I_2(\mathbf{F}) = \frac{1}{2} ((\text{tr } \mathbf{F})^2 - \text{tr}(\mathbf{F}^2)). \quad (4.5)$$

The second invariant criterion was already used in the traditional QC method of atomistic lattices [58], where it quantifies the amount of shear inside the element, ultimately leading to dislocation nucleation/motion (so that the refinement threshold r_0 was related to the Burgers vector). Here, this criterion is chosen as a general measure of the localized deformation.

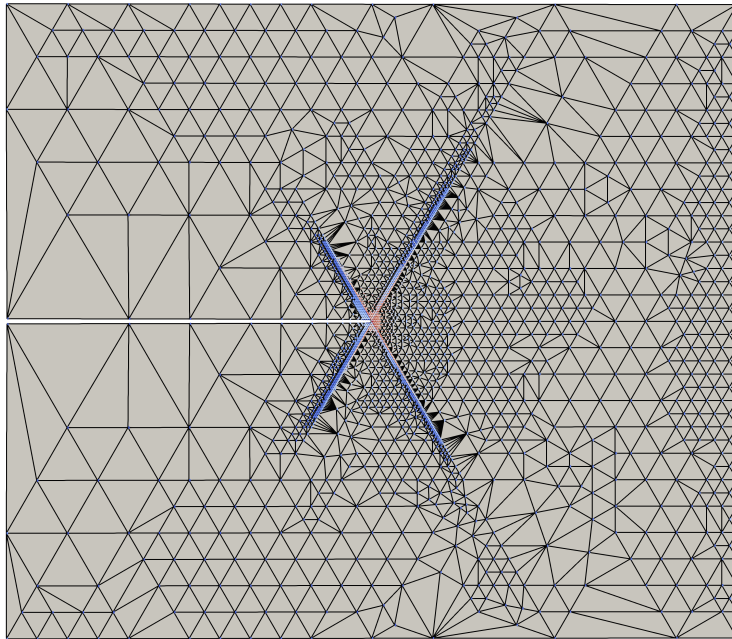
The refinement tolerance is repeatedly decreased, thereby increasing the total number of dofs in the simulation. The maximum tensile stress near the notch tip and the total energy of the system are monitored for convergence of the simulation. Fully-resolved simulations were performed for comparison.

We perform coarse-grained simulations of the triangle, hexagon, kagome, and star-shaped truss lattices undergoing Mode I loading. Following others that have performed boundary layer analysis, we use the elastic constants of Quintana-Alonso et al. [83] for the triangle, kagome, and hexagon lattices, which correspond to a first-order approximation of the homogenized elastic constants in terms of slenderness ratio (i.e. higher-order bending terms are neglected for stretching-dominated lattices). For the star-shaped lattice, which has not been studied before, we use the first-order approximation of the elastic constants which we calculated. For the exact elastic constants for each lattice, see Appendix A.

Figure 4.7 and Fig. 4.8 illustrate the convergence of the maximum tensile stress and of the total energy of the system towards the exact values obtained from fully-resolved simulations. Apparently, the accuracy of the coarse-grained and adaptively refined simulations drastically varies based on lattice topology and relative density; and one generally expects the error to decrease with increasing repUC density. The unrefined *kagome* lattice simulations show large errors in the maximum tensile stress and energy, which are greatly reduced by the adaptive refinement of the mesh while still using only a small fraction of all dofs in the lattice. Furthermore, the refinement procedure fully refines the mesh along the shear bands that emanate from the crack tip in the fully-resolved simulation (see Fig. 4.5). This demonstrates the effectiveness of the adaptive QC method. While the coarse-grained simulation displays significant



(a) Unrefined truss QC mesh.



(b) Adaptively refined truss QC mesh.

Figure 4.5: Coarse-grained kagome lattice in 2D with a pre-existing crack undergoing mode-I loading, simulated by truss QC with adaptive mesh refinement. The colors correspond to the (normalized) maximum tensile stresses in the beams.

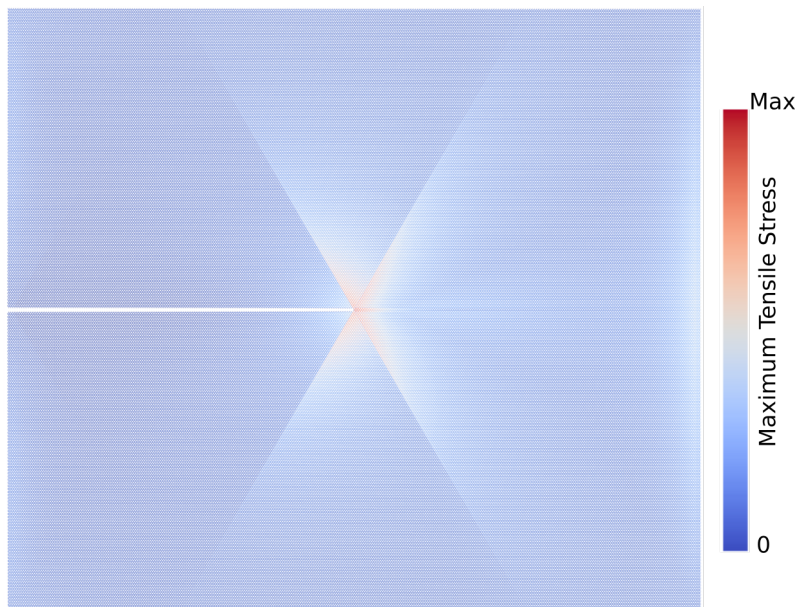


Figure 4.6: Fully-resolved simulation of the 2D kagome lattice undergoing mode 1 loading. The colors correspond to the (normalized) maximum tensile stresses in the beams.

errors (because boundary-layer effects coming from the crack tip interfere with the coarse regions), the refinement technique enlarges the fully-resolved region only where necessary and effectively reduces the error. The *triangle* lattice shows small errors in the energy and maximum stress even in the unrefined case, suggesting that the boundary layer effects from the crack tip are highly local, so that no refinement is needed.

The star and hexagon lattices show considerable errors for the unrefined mesh. Interestingly, while the error decreases as the mesh is refined, the errors do not decrease to reasonably small values as in the previous cases. It is worth noting that the *hexagon* lattice has a bending-dominated Young's modulus ($E \sim \bar{\rho}^3$) and a stretching-dominated bulk modulus ($\kappa \sim \bar{\rho}$), meaning that for low relative densities the effective response is nearly incompressible (see Appendix A). Since we are using linear interpolation on a triangulated mesh, it is possible that we are seeing volumetric locking as in finite elements¹. If this is the case, then the large errors could be attributed to the nearly incompressible nature of the hexagon lattice.

¹In the finite element method, fully-integrated Langrange finite elements (which would be analogous to the QC formulation here) are known to behave overly stiff when simulating nearly incompressible materials, and even lock in the incompressible limit.

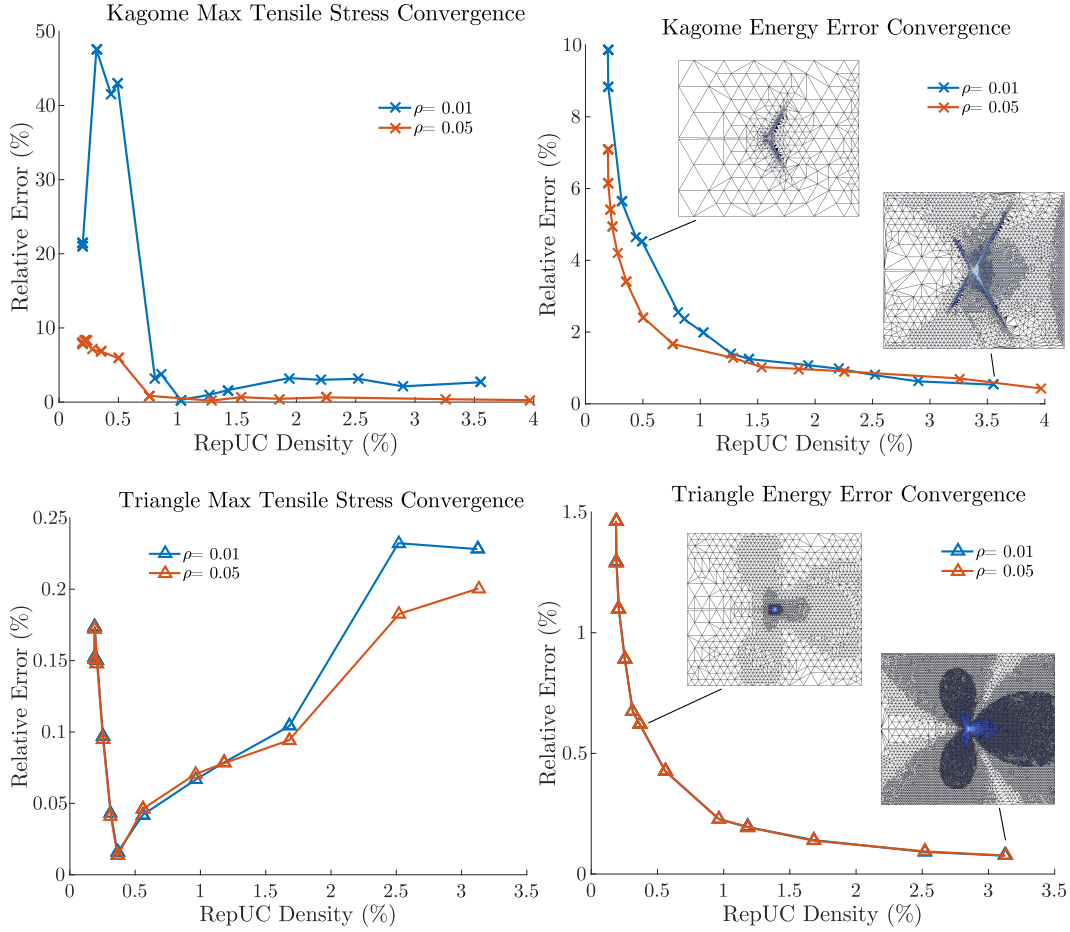


Figure 4.7: Convergence of the maximum strut tensile stress near the notch tip (*left*) and of the total energy (*right*) as compared to the fully-resolved simulation for coarse-grained truss QC fracture simulations of a (*top*) kagome lattice, (*bottom*) triangle lattice in 2D. Insets visualize the adaptively refined QC representation at the indicated repUC densities.

However, the challenges with the star-shaped lattice cannot be explained by volumetric locking, since both bulk and Young moduli of the effective continuum are bending-dominated. On the other hand, when the beams are slender, they act as if they are inextensible, since all macroscopic deformation can be accommodated by the bending of individual beams, which comes with significantly lower strain energy. The fact that we considerably overpredict the energy of the star-shaped lattice (and that the discrepancy increases as we decrease the relative density) suggests that the linear interpolation of the dofs of each individual Bravais lattice does not allow the multilattice to accommodate the macroscopic deformation without the stretching of beams. In other words, if we think of the beams as inextensible members, then the linear interpolation

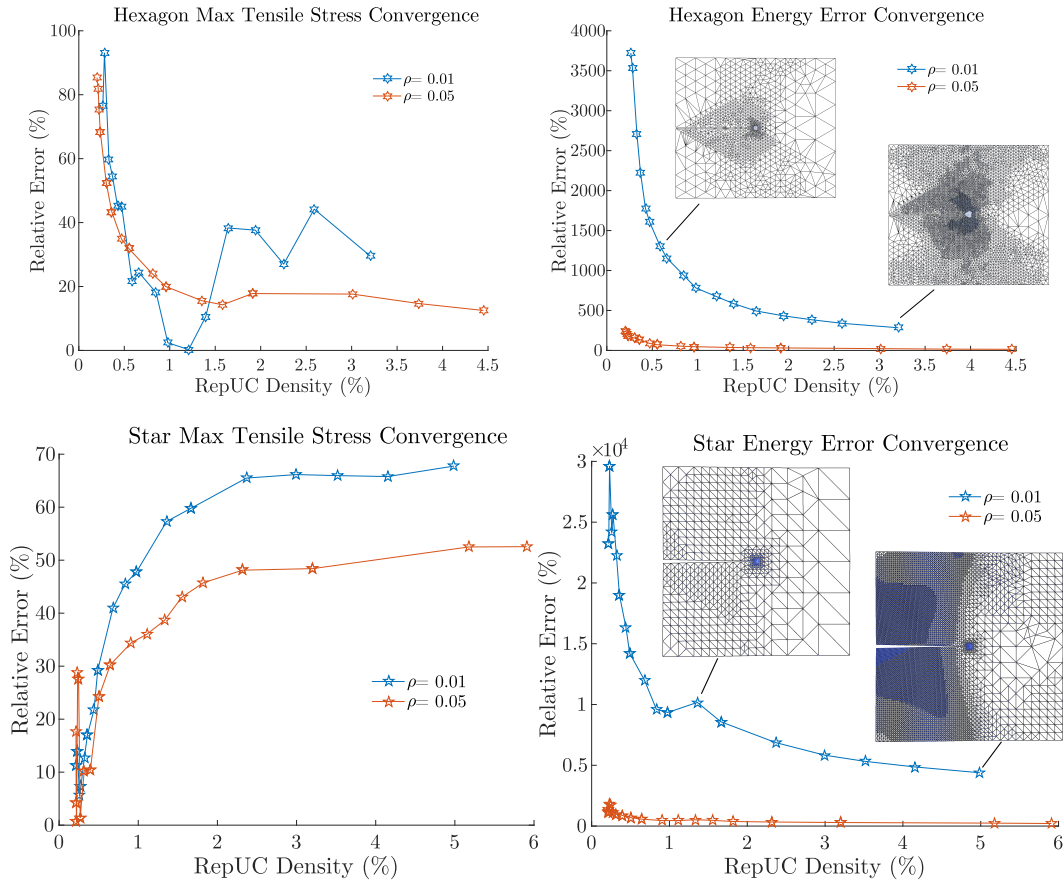


Figure 4.8: Convergence of the maximum strut tensile stress near the notch tip (*left*) and of the total energy (*right*) as compared to the fully-resolved simulation for coarse-grained truss QC fracture simulations of a (*top*) hexagon lattice, (*bottom*) star-shaped lattice in 2D. Insets visualize the adaptively refined QC representation at the indicated repUC densities.

described in Section 3.4.1 leaves the lattice overconstrained for non-uniform deformation (recall that in Section 4.1 we showed that this interpolation performs well for uniform deformation). This overconstraining of the lattice is investigated in more detail in Chapter 5, where we propose and test a solution that mitigates the errors.

It should be noted that the excellent prediction of the elastic constants for the hexagon and star-shaped lattices in Section 4.1 and the overprediction of the energy (and therefore, the stiffness) of the lattices in this section do not contradict each other. The elastic constants simulations showed that the proposed kinematic constraints were able to match the true solution for uniform deformations, while alternative formulations may be preferable for more

complex non-uniform deformations. This is analogous to how fully-integrated standard Lagrangian FE passes the patch test, but can massively overpredict the stiffness of incompressible materials undergoing more complicated deformations.

4.3 Indentation

In order to test the geometric nonlinear capabilities of truss QC, we perform adaptive indentation simulations on 2D and 3D multilattices. Indentation is chosen because it features highly localized deformation, buckling of beams in compression, a non-trivial onset of localization, and it has been used as a standard benchmark in classical QC [59]. We use the nonlinear corotational beams described in Section 3.2.2. The indenter is modeled using an energetic penalty, viz. by applying an external force potential

$$\mathcal{L}_s = \sum_{s=1}^{N_s} w_s \sum_{p \in \Omega_s} \kappa (|\mathbf{x}_p - \mathbf{x}_c| - R)^3, \quad (4.6)$$

where κ is a force constant, and R and \mathbf{x}_c are the radius and center of the spherical indenter, respectively. The QC representation is fully-resolved where there is contact between the truss and the indenter, so the external force represents a penalty for each node in the truss lattice that penetrates the spherical indenter. It should be noted that while the contact between the truss nodes and the indenter is modeled, the contact between individual truss members is not taken into account.

For comparison purposes, the 2D and 3D indentation simulations were repeated with the following settings:

- fully-resolved with corotational beams,
- fully-resolved with linear Euler-Bernoulli beams,
- truss QC with corotational beams, no refinement,
- truss QC with corotational beams, adaptive refinement.

The fully-resolved simulations with corotational beams are the highest fidelity model, and are thus used as the “exact” solution. Simulations with linear beam elements are used to demonstrate the drastic difference in predicted behavior when geometric nonlinearity is taken into account. Lastly, the two coarse-grained simulations are used to demonstrate the accuracy of truss QC and to

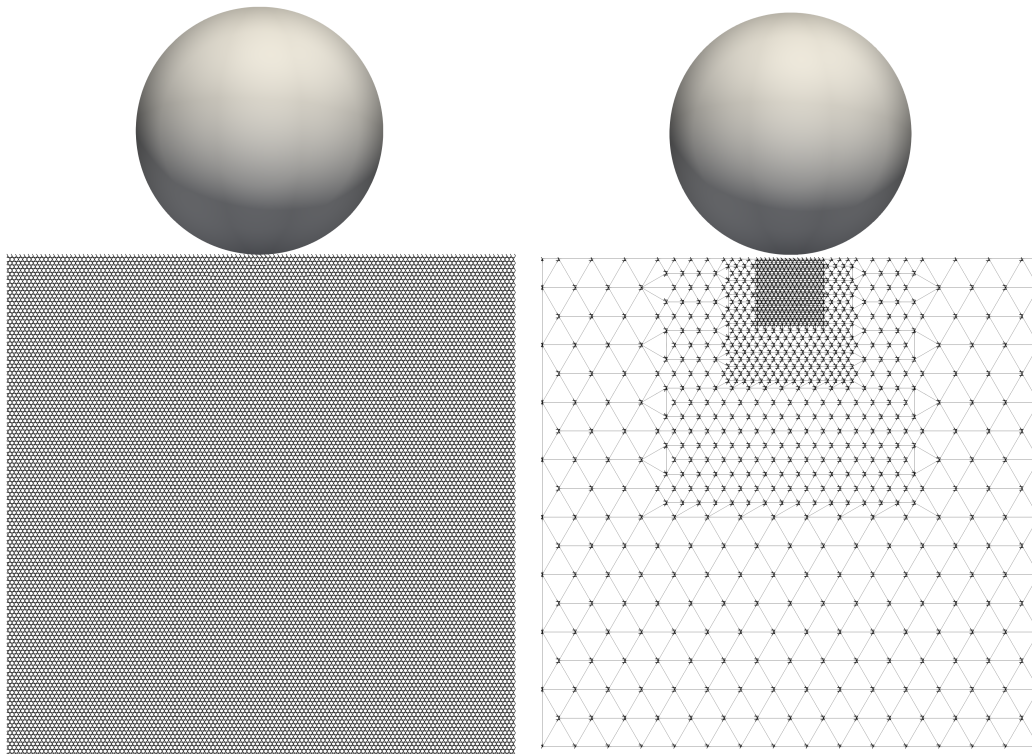


Figure 4.9: Initial geometry for the fully-resolved (*left*) and coarse-grained (*right*) 2D kagome lattice before indentation.

show the importance of adaptive refinement. The adaptive refinement technique of Section 3.5 is used with the same weighting function and refinement criterion used in Section 4.2 (given by Eq. (4.4) and Eq. (4.5)).

In 2D, we simulate indentation into a kagome lattice with relative density $\bar{\rho} = 35\%$ ($t/L \approx 0.20$ where t is the thickness of the truss member), which is tessellated to fill a square of side length $240L$ (see Fig. 4.9). This results in a lattice with roughly 35,000 nodes and 107,000 corotational beam elements. The coarse-grained mesh is created by having an initially fully-resolved region of size $32L \times 32L$ where the indenter makes contact, and gradually coarsening away to fill the domain resulting in a mesh with roughly 1,900 nodes ($\approx 5.3\%$ of the fully-resolved simulation). The indenter has radius $R = 60L$ and indents into the lattice a distance of $20L$ over 50 equal load steps. The refinement algorithm uses a tolerance of $r_0 = 0.075L$.

The obtained load-displacement curve is found in Fig. 4.10. The indenter force (F) is normalized with respect to Young's modulus E and truss length L , while the indenter displacement d is normalized with respect to the truss

length L . Comparing the fully-resolved corotational beam simulations to the linear Euler-Bernoulli beam simulations illustrates that significant geometric nonlinearity occurs during the simulation, justifying the use of corotational beams. Furthermore, the difference between the unrefined and refined truss QC simulations portrays the need for adaptive mesh refinement as the deformation becomes more severe. Most importantly, the refined truss QC simulation shows convincing agreement with the exact solution, even at significant indentation depths. The adaptively refined simulation behaves slightly stiffer than the fully-resolved simulations, which is expected since the interpolation was chosen such that truss QC overpredicts the energy of the exact solution.

Figure 4.10 shows the repUC density of the simulation during refinement along with snapshots of the coarse-grained truss at various load steps. Those snapshots indicate that geometric nonlinearity in the simulation is to a considerable part due to the buckling of truss members underneath the indenter; adaptive refinement expands the fully-resolved region as necessary to capture this localized buckling.

In 3D, a kagome lattice with relative density $\bar{\rho} = 10\%$ ($r/L \approx 0.122$ where r is the radius of the truss member with circular cross-section) is tessellated to fill a cube of side length $35L$ (see Fig. 4.11). This results in a lattice with roughly 44,000 nodes and 130,000 corotational beam elements. The truss QC simulation starts with a mesh containing 1,600 nodes ($\approx 3.5\%$ of the fully-resolved simulation). The indenter has radius $R = 5L$ and indents into the lattice a distance of $6L$ over 50 equal load steps. Refinement uses a tolerance of $r_0 = 0.04L$.

The 3D results – while being similar to those results of the 2D lattice above – demonstrate the power of the 3D truss QC theory and implementation. The normalized load-displacement curve and the adaptive refinement results are shown in Fig. 4.12. Significant nonlinearity is observed in the simulation, and adaptive refinement is necessary and successful in accurately predicting the response of the lattice. The adaptive truss QC simulation agrees well with the exact solution (even though not quite as well as in 2D). However, the main features of the simulation such as the initialization of buckling near the indenter and the general post-buckling behavior match well.

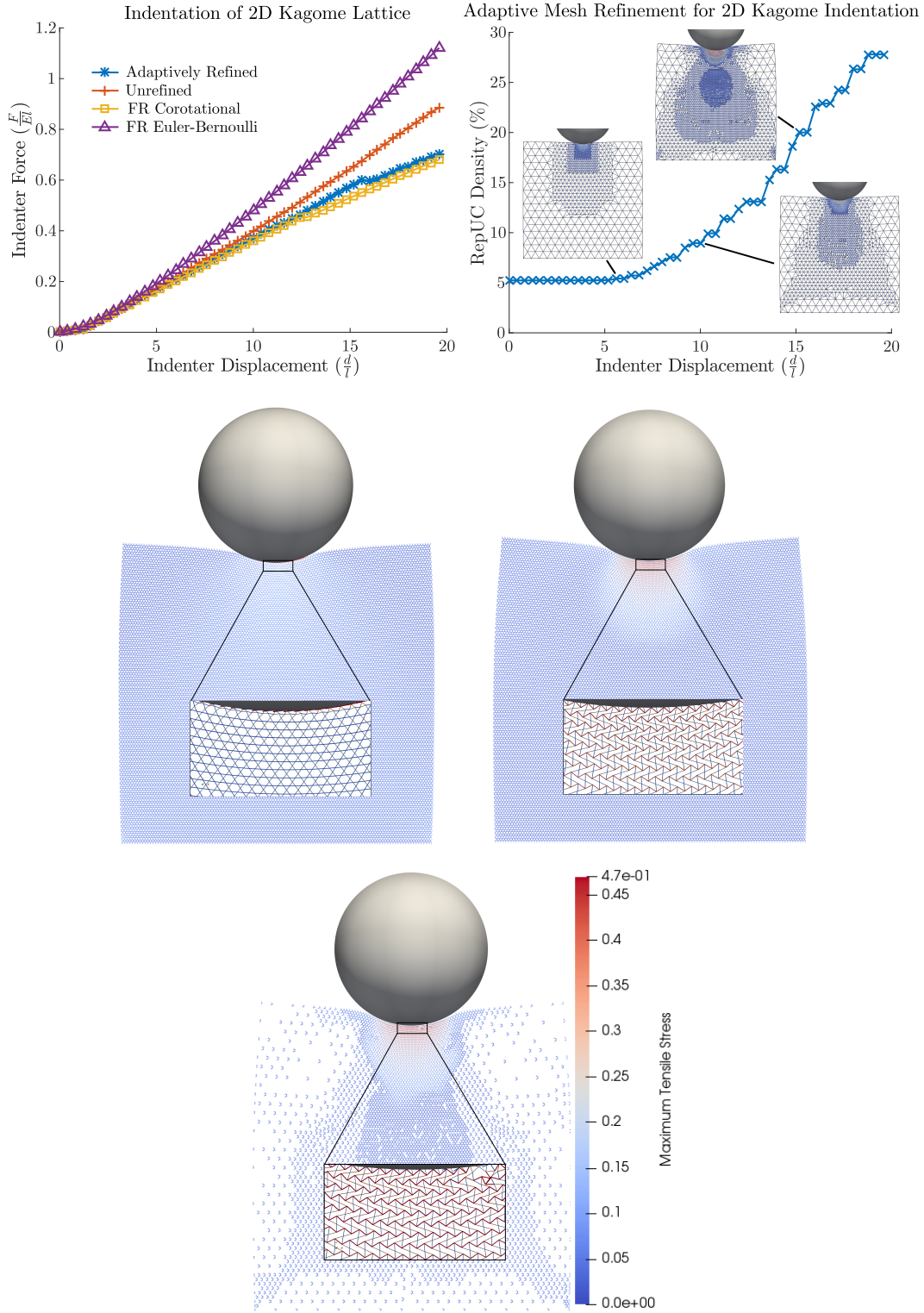


Figure 4.10: (a) Normalized load-displacement curves for the indentation into a 2D kagome lattice, (b) the repUC density as the simulation progresses, and the final deformed configurations of the (c) fully-resolved Euler-Bernoulli beam, (d) fully-resolved corotational beam, and (e) adaptively refined corotational beam simulations. The colors of beams correspond to the maximum tensile stress in each beam normalized by Young's modulus.

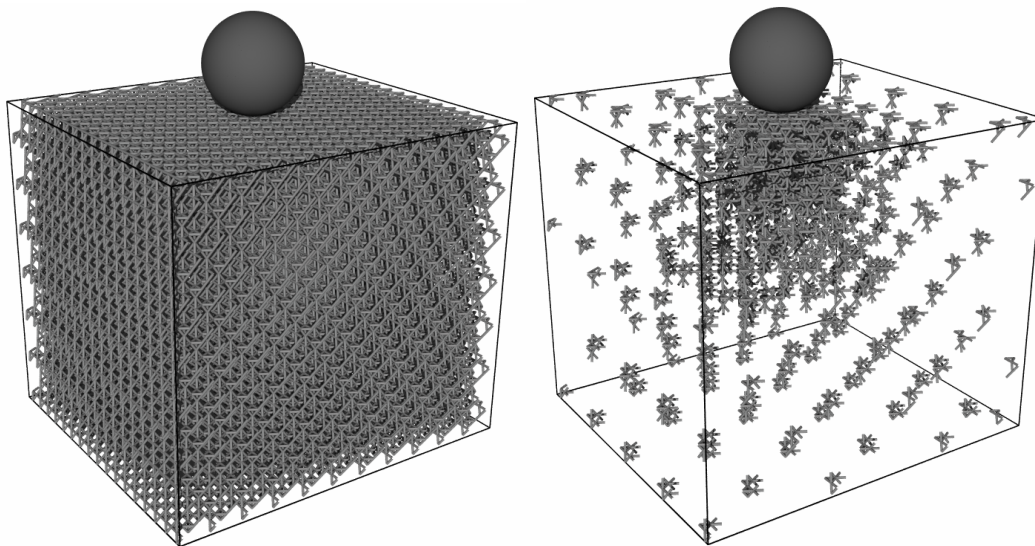


Figure 4.11: **(a)** Initial geometry for the fully-resolved and **(b)** coarse-grained truss QC of a 3D kagome lattice before indentation.

4.4 Dynamic Indentation

In this section, we test our formulation of the QC method modeling history-dependent truss lattices. We perform dynamic simulations of a 2D triangle truss lattice subject to a cylindrical indenter moving at a constant velocity². The geometry of the truss lattice is a 192×152 rectangle, with each truss having a length $L = 1$ (see Fig. 4.13). The indenter is taken to have a radius of $r = 30$, and moves a distance $d = 10$ into the lattice over a time period of $t = 200$. We use the variational formulation of the Newmark method outlined in Section 3.3.2 with constants $\beta = 1/4$ and $\gamma = 1/2$ (corresponding to the second-order accurate average acceleration method) and a time step of $\Delta t = 1$. To model the individual truss members, we use the corotational beams described in Section 3.2.2 paired with the brittle failure model in Section 3.3.1 using the average tensile stress (ATS) as the yield criterion. The corotational

²This is equivalent to modeling the impact of a cylinder with infinite mass.

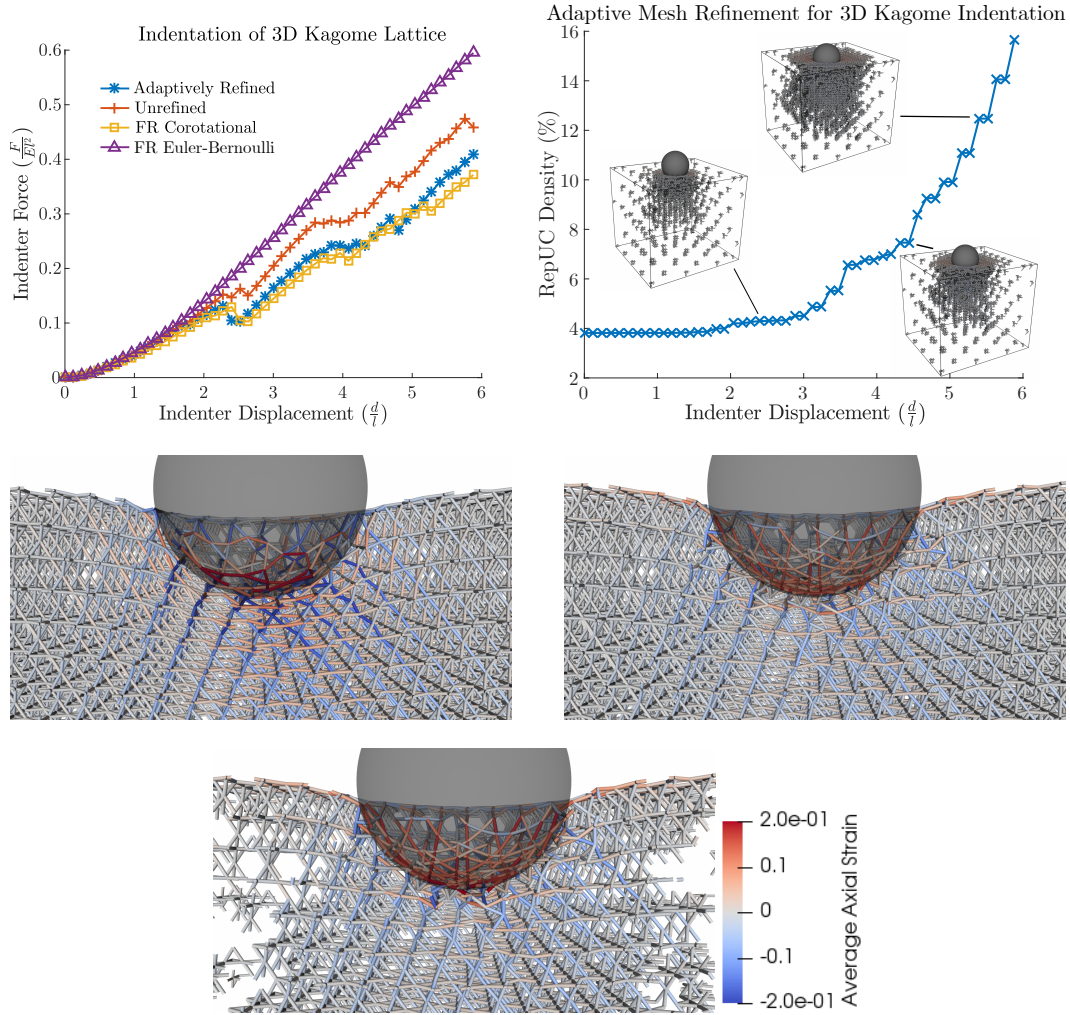


Figure 4.12: Normalized load-displacement curve for the indentation of a 3D kagome lattice (*top left*), the repUC density as the simulation progresses (*top right*), and the slices of the final deformed configurations of the fully-resolved (*bottom left*) and adaptively refined (*bottom right*) simulations with average axial strain contours.

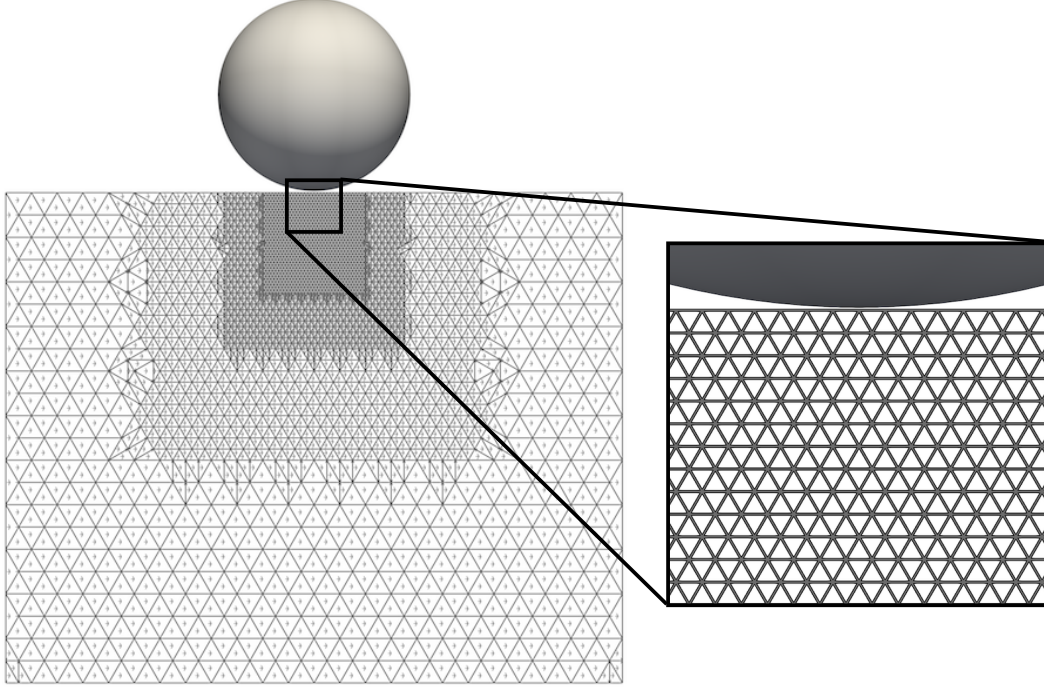


Figure 4.13: Geometry of the coarse-grained dynamic indentation simulation. The left shows the initial coarse-grained truss lattice with the mesh and sample unit cells.

beams are taken to have the lumped mass matrix

$$\mathbf{M} = \frac{\rho AL}{2} \begin{bmatrix} 1 & & & \\ & 1 & & 0 \\ & & \frac{L^2}{12} & \\ & 0 & & 1 \\ & & & & \frac{L^2}{12} \end{bmatrix}, \quad (4.7)$$

where A and L are the cross-sectional area and beam length respectively. The triangle lattice is taken to have a relative density $\bar{\rho} = 10\%$, and the constituent material is given a density of $\rho = 1$. The Young's modulus of the constituent material is $E = 10$, and the yield strength used in the failure criterion is $\sigma_y = 0.3$. We use the process of solving the elastic system and failing one beam at a time described in Algorithm 1.

The problem was simulated using the QC method with the first-order summation rule described in Section 3.4.2. The mesh was adaptively refined using the general refinement framework outlined in Section 3.5, with the weighting

function

$$F(K) = L_e(K), \quad (4.8)$$

where $L_e(K)$ is the length of the longest edge of K , and the refinement metric

$$f(\mathbf{F}) = \max_{\|\hat{\mathbf{X}}\|=1} \|\mathbf{F}\hat{\mathbf{X}}\| - 1 \quad (4.9)$$

was used to mark elements for refinement. Equation (4.9) represents the maximum possible axial strain of any truss member inside the macroscopic element. In order to assure that failure of beams only occurs in fully-resolved regions, we use a refinement threshold of $r_0 = 0.05$.

Snapshots of the adaptively refined coarse-grained simulation are compared with the fully-resolved simulation in Fig. 4.14. It can be seen that the overall behavior of the truss is the same in both simulations. In both cases, a region underneath the indenter catastrophically fails, and a long crack propagates through the truss lattice. We note that the onset of failure in the coarse-grained simulation occurs in the exact same location and only two load steps earlier than the fully resolved simulation. The fact that the exact failure pattern does not match in the two simulations is expected; the process of failing beams one at a time breaks the symmetry in the problem, and even numerical noise could result in different crack propagation patterns (similar to how exact dislocation motion in atomistics often is not repeatable).

From Fig. 4.14, it is clear that the propagation of the crack in both the QC and the fully-resolved simulations move at the same speed. The average rupture speed of the longest crack (calculated by computing the distance and time it took to travel from initiation to interacting with the bottom boundary) in both simulations was found to be $v_r \approx 1.59$. Using the homogenized elastic constants and average density of the truss lattice, the longitudinal and shear wave speeds of the effective material are

$$c_l = \sqrt{\frac{E^*(1 - \nu^*)}{(1 + \nu^*)(1 - 2\nu^*)\rho^*}} = \sqrt{5} \quad c_s = \sqrt{\frac{\mu^*}{\rho^*}} = \frac{\sqrt{5}}{2}. \quad (4.10)$$

Interestingly, the rupture speed is somewhere in between the shear and longitudinal wave speed of the homogeneous material. For mode I crack propagation, the maximum rupture speed of a crack is the Rayleigh wave speed, $c_R < c_s$ [43]. However, the greatest mode II rupture speed predicted by dynamic fracture mechanics is $v_r^{\max} = \sqrt{2}c_s \approx 1.58$ [42], which is very close to (and within

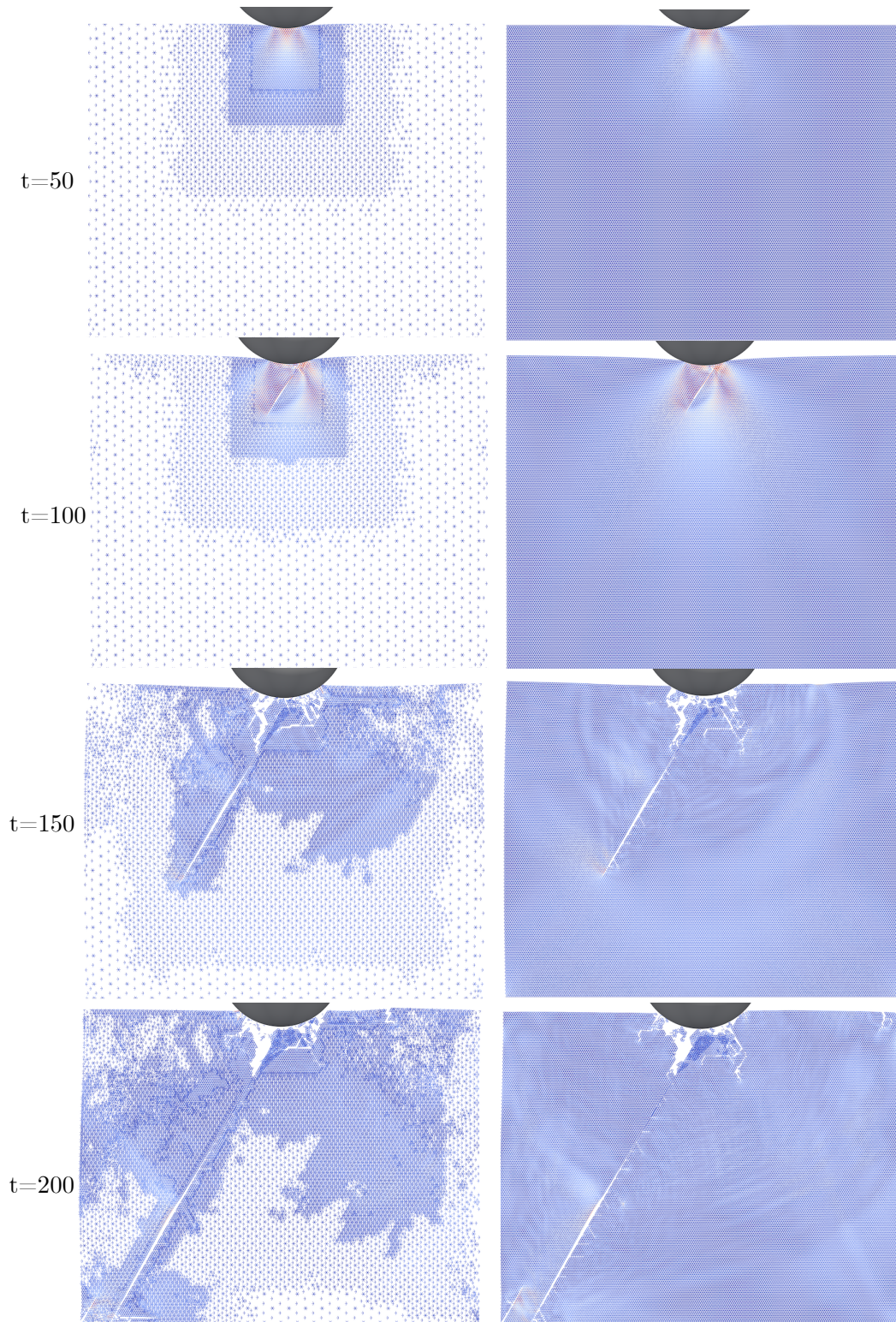


Figure 4.14: Snapshots of the truss lattice: coarse-grained (left) and fully-resolved (right). Colors correspond to the maximum stress in each beam, and the broken beams are not shown.

the approximation error of) the computed rupture speed of $v_r \approx 1.59$. In the fracture mechanics field, this phenomenon is called *supershear* rupture since the crack travels faster than the shear wave speed, and has been observed both in the lab [90] and in nature during earthquakes [24].

While the overall behavior of the lattice matches in both cases, there are significant discrepancies in the way elastic waves behave between the two simulations. Figure 4.15 consists of the deformed meshes colored by the magnitude of the acceleration at various pre-rupture times to show how the elastic waves created by the indenter propagate throughout the coarse-grained and fully-resolved truss. The fully-resolved simulations show the longitudinal waves propagating throughout the body along with Rayleigh waves confined near the upper boundary (and some reverberations from the truss bouncing off the indenter). Initially, when the elastic waves are contained in the fully-resolved region, the coarse-grained simulation expectedly matches the fully-resolved simulation. However, when the elastic waves reach the boundaries of the coarser mesh, the high-frequency components of the waves are artificially reflected back into the fine domain. By the time the elastic waves reach the coarsest region of the domain, the waves have dispersed to the point where they are no longer recognizable.

The fact that the elastic waves bounce off mesh boundaries is not unique to the QC method; the same effect has been known to occur in dynamic finite element simulations with non-uniform meshes [13, 14]. Although not seen in the simulations here, wave reflections could potentially trap energy in fully-resolved regions and cause excess damage. While there has been many proposed ways to deal with this spurious reflection, it is not a focus of this thesis and is left for future work.

4.5 Hole in Plate Made of an Elasto-Plastic Truss Lattice

We proceed to assess the performance and accuracy of the proposed QC method applied to three dimensional truss lattices undergoing inelastic deformations. The test case is the classic cylindrical hole in a large plate being stretched in-plane. The specimen (see Fig. 4.16), is an octet truss lattice plate with a unit cell size of 1 (resulting in truss lengths $L = \sqrt{2}/2$) with a thickness of 9 unit cells, and width and length of 96 unit cells in each direction. All unit cells within a radius of 8 units from the center are removed from the geometry

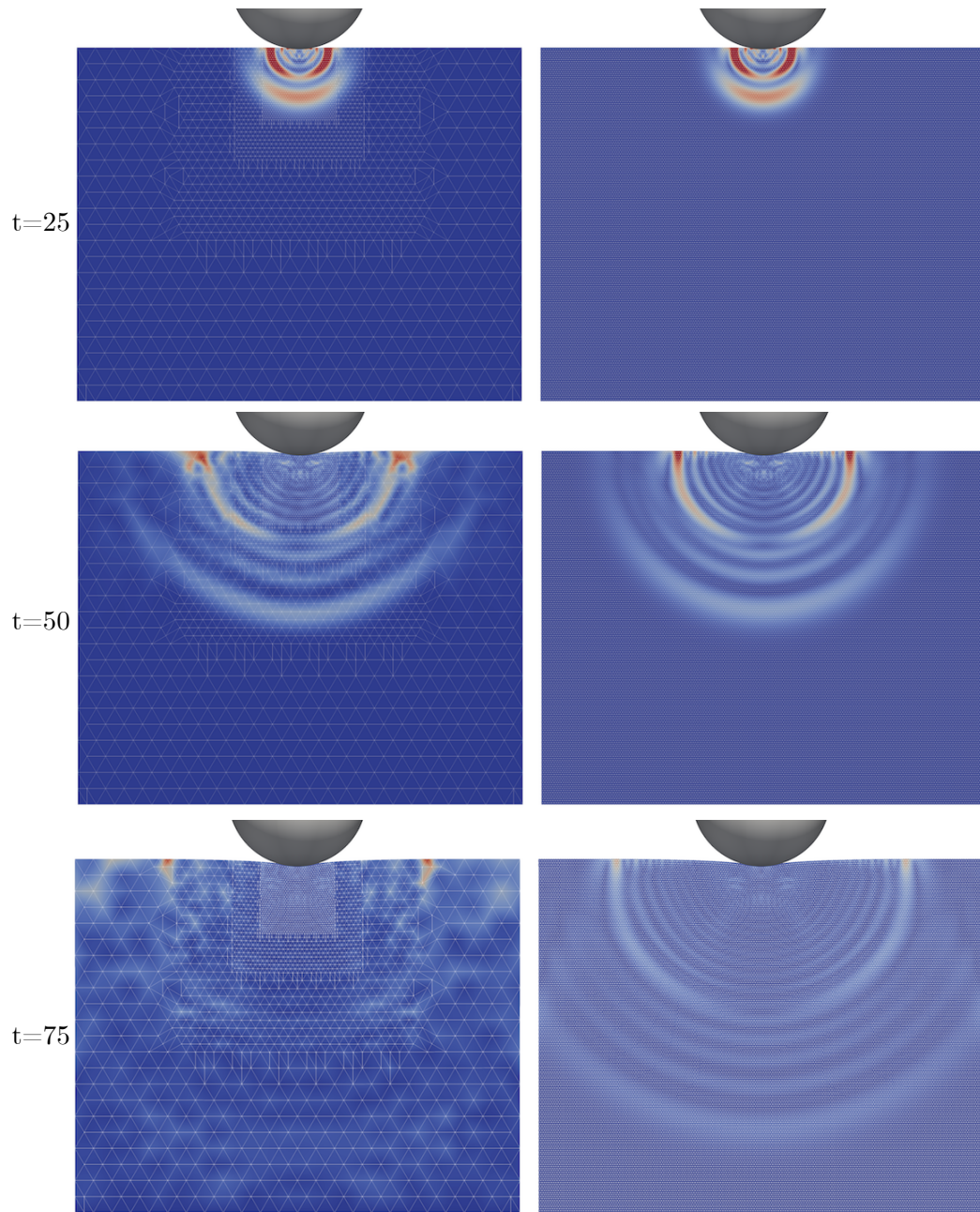


Figure 4.15: Magnitude of the acceleration in the coarse-grained (left) and fully-resolved (right) simulations at various time steps before failure of the truss lattice. The elastic waves can be seen artificially bounding off mesh boundaries in the coarse-grained simulations.

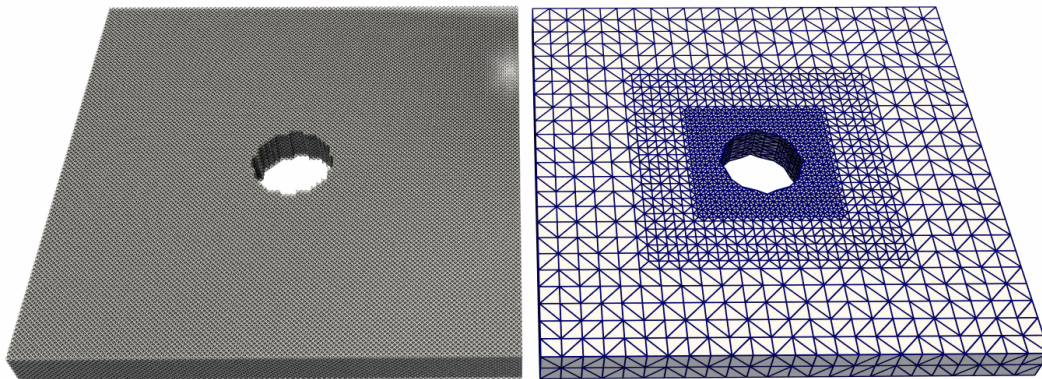


Figure 4.16: Cylindrical hole in octet lattice plate. The fully-resolved truss lattice (left) and the coarse-grained mesh (right)

to create the cylindrical hole³. The resulting geometry is a plate that has roughly 100,000 unit cells, 400,000 nodes, and 2,000,000 truss members.

Since the octet lattice is a rigid, stretching-dominated lattice, we model the individual truss members with large rotations bars, using the isotropic hardening plasticity model described in Section 3.3.1. The constituent material is taken to have a linear elastic regime with Young's modulus $E = 10$, and initial yield stress of $\sigma_y = 2$ with a hardening constant of $K = 10/99$ such that the plastic modulus is $H = 1$. The truss has a relative density of $\bar{\rho} = 10\%$, and the plate undergoes a macroscopic strain of 20% over 20 equal load steps.

The repUCs were chosen by creating a fully resolved $30 \times 30 \times 9$ unit cell block around the hole, then uniformly coarsening away from the hole to reach the specimen dimensions of $96 \times 96 \times 9$. The coarse-grained simulation by comparison consisted of roughly 10,500 repUCs, and 42,000 nodes. The coarse-grained simulations were performed using both the first-order and second-order summation rules described in Section 3.4.2. Figure 4.17 shows the macroscopic stress-strain relation (where the stress is taken as the average force at the ends normalized by the area) predicted by the coarse-grained and fully-resolved simulations.

It is clear from the figure that the second-order summation rule matches the fully-resolved simulation better than the first-order rule. This is because the plate is only 9 unit cells thick, and as mentioned in Section 3.4.2, the first-

³The hole that is created is not exactly cylindrical. Since the full cubic unit cells are removed from the geometry, the hole has jagged edges and only approximates a cylindrical hole.

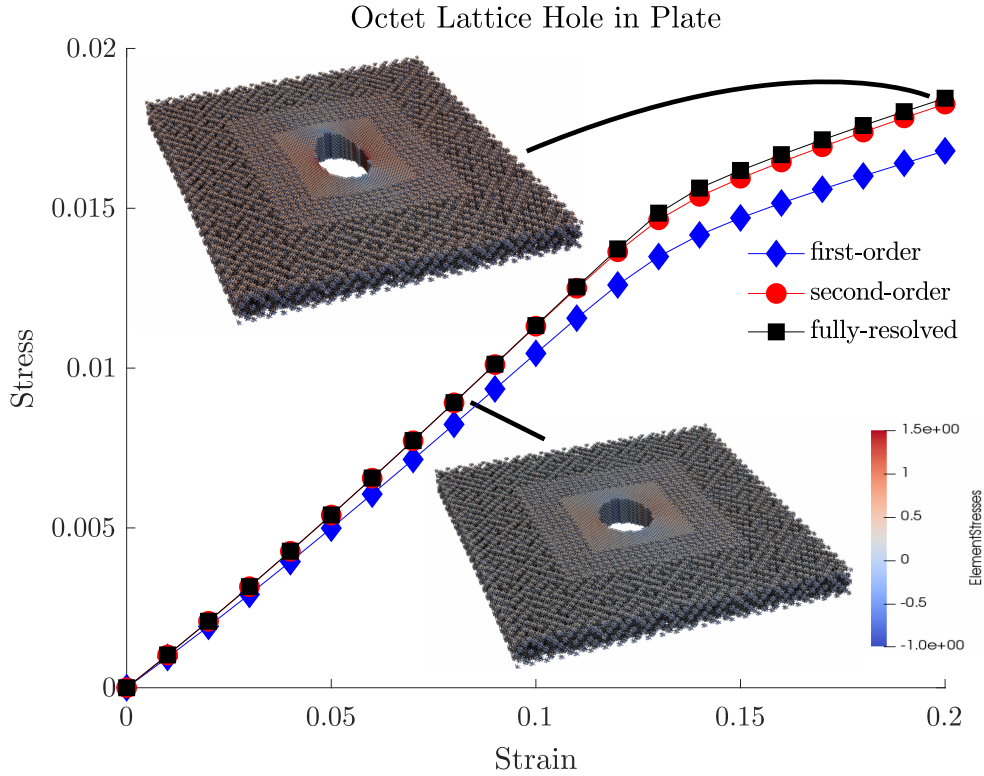
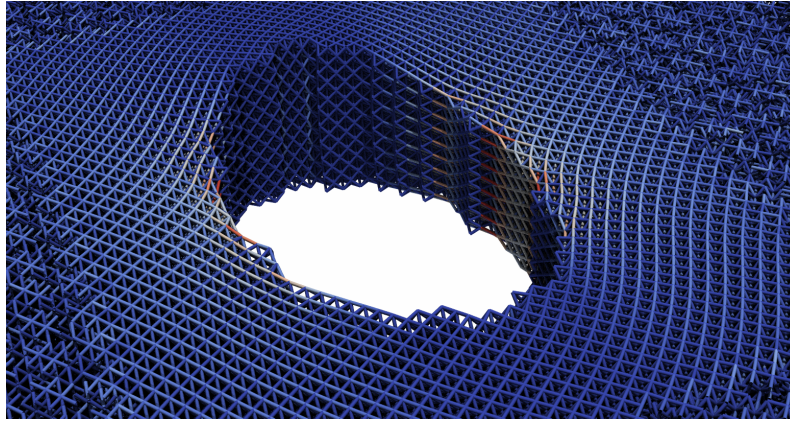


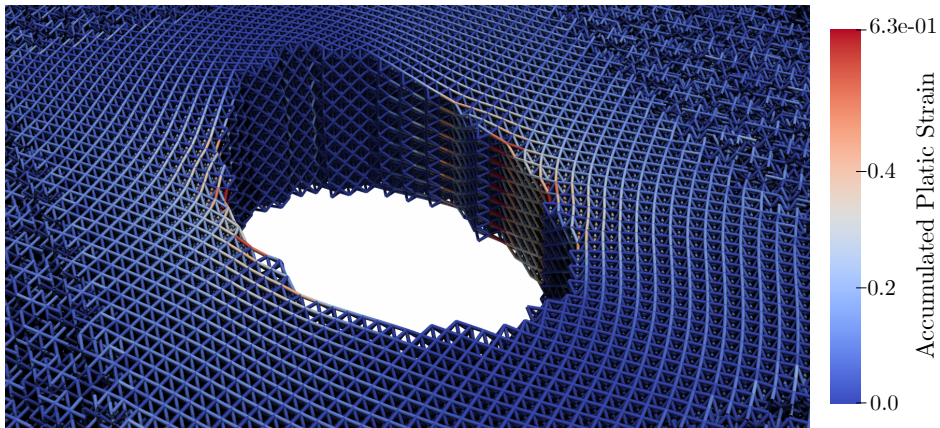
Figure 4.17: Average stress-strain relation for the octet truss hole-in-plate extension simulations with first- and second-order summation rules, compared with the fully-resolved simulation. The insets are colored by the axial stress in each bar in the specimen at macroscopic strain values of $\varepsilon = 0.08$ and $\varepsilon = 0.20$.

order summation rule poorly approximates the number of unit cells in thin geometries (i.e. where there are many boundary unit cells compared with the total number of unit cells). On the other hand, the second-order summation rule, which does a better job at approximating the number of unit cells that lie on the boundary of the geometry, matches the fully-resolved simulation almost perfectly.

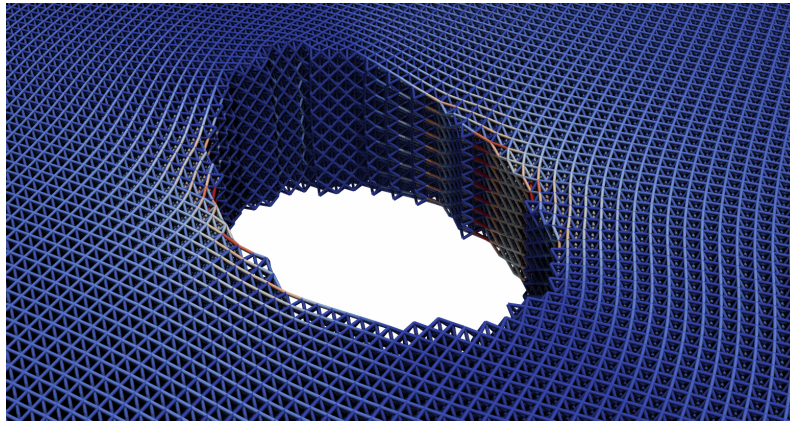
As an additional comparison, Fig. 4.18 shows the equivalent plastic strains in the bars near the hole in the fully-resolved and coarse-grained simulations. Even though the QC method uses roughly 1/10 of the dofs of the fully-resolved simulation, the plastic strains near the hole match the fully-resolved simulation well. This is another example of where the QC method is powerful: the global quantities (i.e. average stress) and the stress concentrations and other local quantities are accurately captured in regions of interest using a fraction of the dofs needed for the fully-resolved simulation.



(a) Coarse-grained with first-order summation rule.



(b) Coarse-grained with second-order summation rule



(c) Fully-resolved

Figure 4.18: Plastic strains near the cylindrical hole at 20% macroscopic strain for the coarse-grained simulations with first-order summation rule (*top*), second-order summation rule (*middle*), and the fully-resolved simulation (*bottom*). The maximum plastic strains are 0.597 (coarse-grained, first-order summation) 0.632 (coarse-grained, second-order summation), and 0.627 (fully-resolved).

HIGHER-ORDER QUASICONTINUUM TECHNIQUES AND STRETCH LOCKING

As seen in the fracture toughness simulations in Section 4.2, the QC method developed in Chapter 3 can display inaccuracies when modeling the response of bending-dominated multilattices. The inability of the method to model bending-dominated lattices is theorized to occur because the kinematic constraints overconstrain the lattice such that the bending-dominated lattice cannot undergo inhomogeneous deformations without stretching individual beams. Therefore, it is desirable to extend the QC method using higher-order interpolation (and corresponding higher-order summation rules) towards improving the predicted response of bending-dominated multilattices. Before formulating the higher-order method in Section 5.2, we take a closer look at the locking phenomenon in Section 5.1 by using a test case common in finite elements. Finally, we test the newly-developed method in Section 5.3 in simulations where locking is known to occur.

5.1 Stretch Locking

When extending the existing Bravais lattice QC method for trusses to multilattices, some of the bending-dominated lattices were seen to exhibit a locking behavior that is similar to volumetric locking in finite elements (see Section 4.2). In the finite element method, volumetric locking occurs in all fully-integrated Lagrange finite elements when modeling (nearly) incompressible materials. It occurs because the piecewise polynomial interpolation of displacements is not a sufficiently rich basis to approximate the (nearly) deviatoric strains of the true solution.

In the QC method for trusses, locking is not due to the near-incompressibility of a lattice (e.g. the hexagonal lattice), since it is also seen in lattices topologies where all deformation modes of the homogenized material are bending-dominated (e.g. the star-shaped lattice). Therefore, the locking is theorized to occur because of the near-inextensibility of the truss members (i.e. the axial stiffness is orders of magnitudes greater than the bending stiffness) for bending-dominated lattices, and thus, we refer to this locking as *stretch locking*. Here

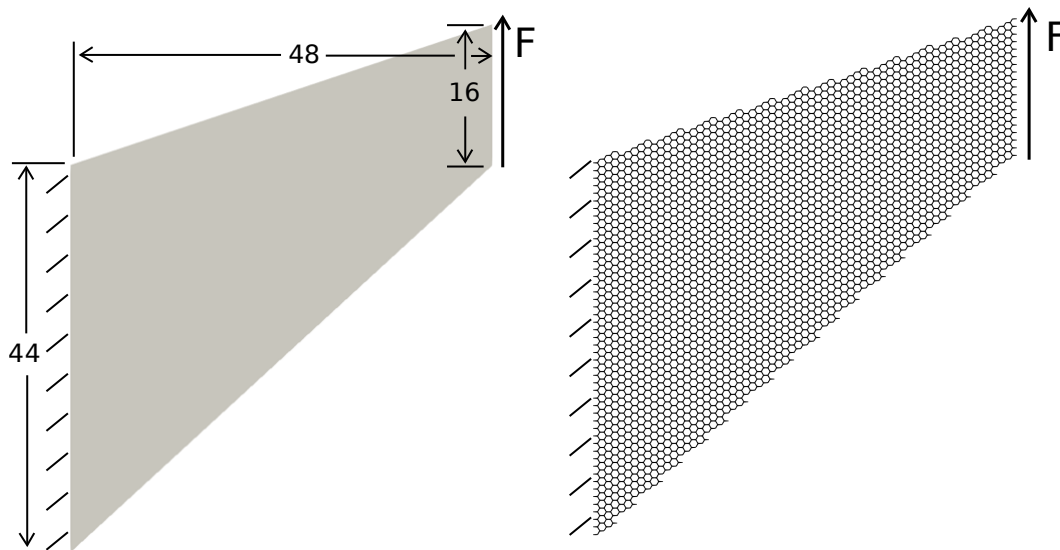


Figure 5.1: Traditional Cook's membrane geometry (*left*) and the corresponding hexagon lattice membrane (*right*) with each truss member having a length of $L = 1$.

we investigate the truss lattice equivalent of Cook's membrane problem, which is a well-known test case for volumetric locking in finite elements.

Cook's membrane consists of the geometry shown in Fig. 5.1 with the left side of the membrane held fixed, and a distributed load placed on the right side of the membrane. The truss lattice equivalent problem is constructed by filling Cook's membrane geometry with a given truss topology with specified truss length L (see Fig. 5.1). On the left side of the membrane, one node in each unit cell is held fixed, and similarly on the right side of the membrane; a vertical external force is placed on one node in each unit cell on the boundary.

We perform truss lattice versions of Cook's membrane problem using the QC method outlined in Chapter 3 with several two-dimensional lattice topologies: square, triangle, hexagon, kagome, and star-shaped. Each lattice is scaled such that the length of an individual beam is $L = 1/10$, and the beam was modeled using linearized Euler-Bernoulli beams described in Section 3.2.2. The fully-resolved problem is simulated and compared with coarse-grained simulations where the mesh is coarsened by choosing every n^{th} unit cell as a repUC, $n = \{2, 3, 4, 6, 8\}$ (see Fig. 5.2)¹. In order to monitor the locking

¹Since this is a 2D simulation, choosing every n^{th} unit cell results in a simulation with an approximate repUC density of $1/n^2$.

phenomenon, the strain energy of the lattice is used as a metric².

Figure 5.3 shows the strain energy (normalized with respect to the fully-resolved strain energy) of the coarse-grained simulations for the various truss lattice topologies with relative density $\bar{\rho} = 1\%$. The square, triangle, and kagome lattices do not exhibit stretch locking; as the repUC density is decreased, the strain energy in the simulation closely approximates the strain energy in the fully-resolved simulation. However, it is clear that the hexagon and star-shaped lattices — the only two bending-dominated multilattice topologies — do exhibit locking since the normalized strain energy immediately drops upon coarsening. The fact that this happens for both the hexagon (nearly incompressible homogenized effective material) and the star-shaped (all modes bending-dominated) lattices signifies that this is not due to volumetric locking like in finite elements, but to stretch locking.

Figure 5.4 shows how changing the relative density of the truss lattices effects the locking phenomenon. In both lattices, as the relative density increases (and the difference between stretching and bending stiffness decreases), the locking becomes less severe. This is further evidence that the locking occurs because the interpolation of the dofs over-constrains the system such that bending-dominated lattices cannot undergo inhomogeneous deformation without stretching individual beams.

5.2 Higher-Order Interpolation and Summation Rules

Most implementations of the QC method resort to using linear interpolation of the dofs in coarse-grained regions. However, there have been some formulations using alternative interpolations such as meshless methods [60] and quadratic and/or cubic Hermite interpolation [16, 114] to model 2D plate-like lattices. In particular, the higher order QC method of Beex et al. [16] is relevant because the authors used cubic interpolation of translational dofs and quadratic interpolation of rotational dofs in coarse-grained areas to model in- and out-of-plane bending of a 2D lattice plate. The method in [16], inspired by plate finite element shape functions, was developed specifically to model the

²Since the problem is linear elastic, the total strain energy at equilibrium is equal to half the work done by the external forces, $1/2 \mathbf{F} \cdot \mathbf{u}$ on the right hand side of the membrane due to Clapeyron's theorem [65]. The force on the right hand side is constant and in the vertical direction, so comparing the strain energy of the lattice is equivalent to comparing the average vertical displacements on the right hand side.

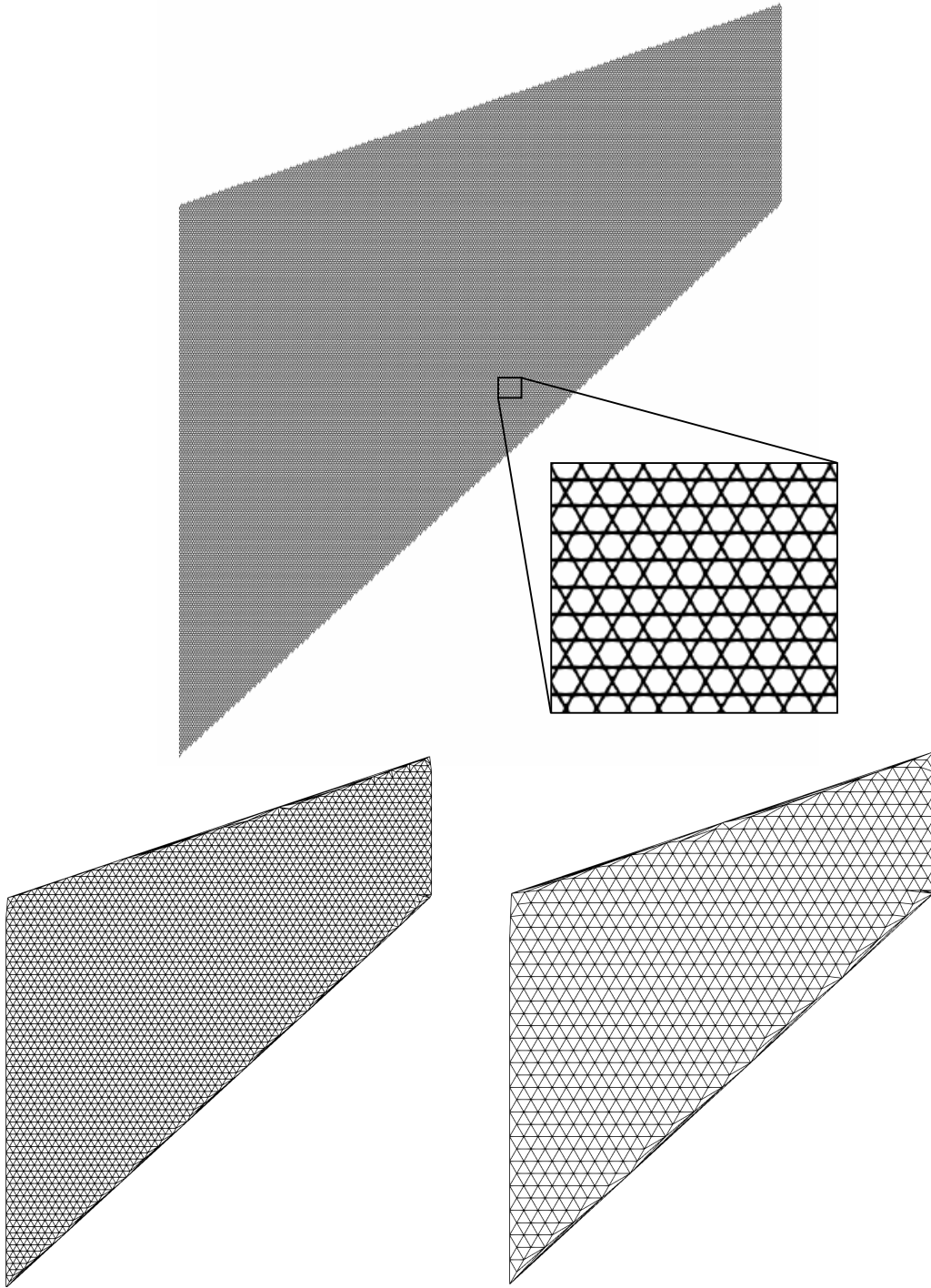


Figure 5.2: Example meshes for the coarse-grained truss lattice Cook's membrane simulations. Full resolution (*top*), coarsening of $n = 4$ (*bottom left*), and coarsening of $n = 8$ (*bottom right*) of the 2D kagome lattice.

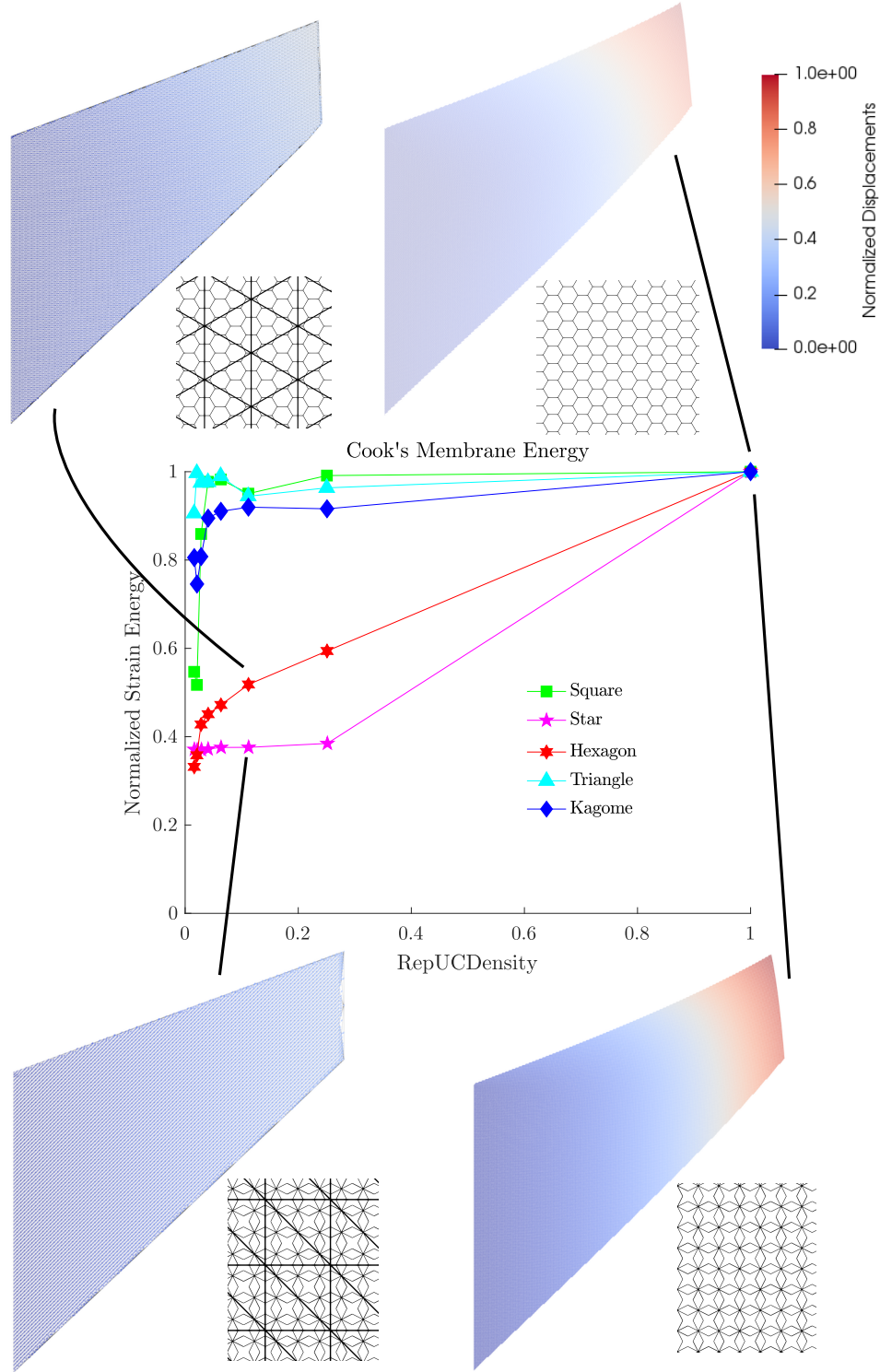


Figure 5.3: Normalized strain energy of truss lattice versions of Cook's membrane problem for various 2D truss topologies with relative density $\bar{\rho} = 1\%$. Insets show the deformed fully-resolved and coarse-grained (with repUC density of $1/9$) simulations of the hexagon (*top*) and star-shaped (*bottom*) lattices, colored by the displacements normalized by the maximum displacement of the fully-resolved truss.

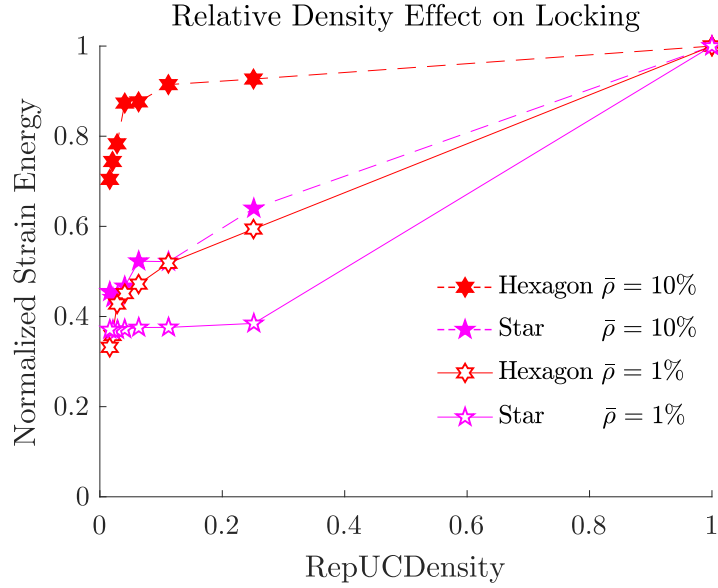


Figure 5.4: Strain energy of the hexagon and star-shaped lattices at relative densities of $\bar{\rho} = 10\%$ and $\bar{\rho} = 1\%$.

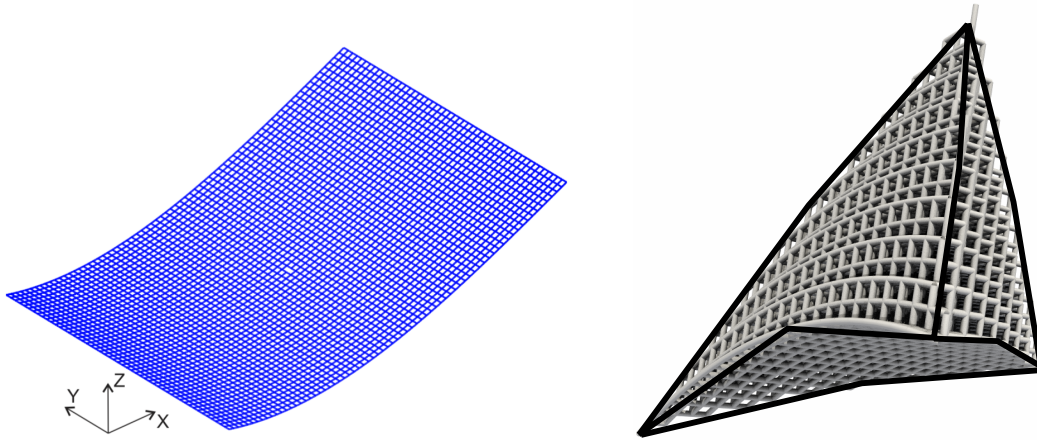


Figure 5.5: An example of higher-order interpolation QC used to model bending of planar lattices (*left*, reprinted from [16] with permission of the copyright holder, Elsevier), and an example of a quadratic tetrahedron in the QC method used to model a "lattice continuum" (*right*).

bending of a "lattice plate" rather than a "lattice continuum". This chapter is devoted to developing a higher-order QC method for the latter (see Fig. 5.5).

It should also be noted that in [16], the higher-order QC technique had a number of undesirable properties which the technique outlined here will not share. In order to model out-of-plane bending, the translational and rotational dofs of the lattice were interpolated using cubic and quadratic shape functions, respec-

tively, leading to different representative sites for translational and rotational dofs (and, therefore, different meshes). In addition, the representative sites did not lie on valid lattice sites³. Furthermore, the kinematic constraints that were introduced resulted in a non-conforming triangulation at the boundary of the fully-resolved region and the coarse-grained region. The higher-order QC method outlined in this chapter will be shown to result in a *conforming* mesh throughout the entire domain with identical representative sites for all dofs.

5.2.1 Kinematics

Just like the QC formulation with affine shape functions are analogous to linear triangles and tetrahedra in finite elements, it is desired to create higher-order QC elements using the interpolation functions of higher-order simplicial elements in finite elements. In order to do so, we need to take some care in choosing repUCs to create higher-order meshes while maintaining the fact that all repUCs lie on valid lattice sites.

The task of choosing repUCs can be rephrased by choosing locations in the domain with integer Bravais coordinates, i.e. choosing points from the set

$$\left\{ \mathbf{X} \mid \mathbf{X} \in \Omega \text{ and } \mathbf{X}_{\text{rep}} = \sum_{i=1}^d c_i \mathbf{a}_i + \mathbf{S}_0, \ c_i \in \mathbb{Z} \right\}, \quad (5.1)$$

where \mathbf{S}_0 is some global offset vector. In order to construct a conforming mesh of second-order Lagrange finite elements⁴ where all repUCs lie on valid lattice sites, we first select unit cells that lie on sites with Bravais coordinates $c_i/2 \in \mathbb{Z}$. In other words, the vertex repUCs would be placed on sites with even Bravais coordinates. These lattice sites are used to create a first-order mesh. Then, the repUCs corresponding to the mid-edge nodes of the quadratic Lagrange finite element are placed in the mesh. The locations of these repUCs come from an affine mapping of the parent Lagrange element, and have the locations

$$\left\{ \mathbf{X} \mid \mathbf{X} = \mathbf{X}_j + \frac{1}{2}(\mathbf{X}_j - \mathbf{X}_k), \ \mathbf{X}_j, \mathbf{X}_k \in K_v, \right\}, \quad (5.2)$$

³While there is nothing intrinsically wrong about not having representative sites lie on valid lattice sites, it is not desirable since the sites lose their physical significance, and can cause problems if paired with automatic refinement (e.g. zero energy modes appearing due to more representative sites than underlying lattice sites).

⁴While the formulation outlined here can be used for interpolation of an arbitrary degree, we restrict ourselves here to quadratic interpolation.

where K_v are the vertex repUCs of element K . When writing out the vertex repUC locations in terms of their Bravais coordinates $c_{i,j}$ and $c_{i,k}$, i.e.

$$\mathbf{X} = \sum_{i=1}^d c_{i,j} \mathbf{a}_i + \frac{1}{2}(c_{i,j} - c_{i,k}) \mathbf{a}_i = \sum_{i=1}^d \left(c_{i,j} - \frac{c_{i,k}}{2} \right) \mathbf{a}_i, \quad (5.3)$$

it is clear that the newly added repUCs also lie on valid lattice sites. The vertex repUCs were chosen such that $c_{i,k}/2 \in \mathbb{Z}$, so the Bravais coordinates of the new sites $(c_{i,j} - c_{i,k}/2) \in \mathbb{Z}$ as well.

Once all of the vertex and mid-edge repUCs are created, the kinematic interpolation is given by

$$\tilde{\varphi}(\mathbf{X}_u) = \sum_{r=1}^{N_{\text{rep}}} N_r(\mathbf{X}_u) \tilde{\varphi}_r, \quad (5.4)$$

where $N_r(\mathbf{X}_u)$ are quadratic Lagrangian shape functions.

It is worth noting here that, in the fully-resolved region, it makes no difference whether quadratic or linear interpolation is used. Since every unit cell is a repUC and the energy is computed exactly in these regions, the shape functions are only evaluated at the repUC locations. If linear elements are used adjacent to quadratic elements in the fully-resolved region, the nonconformity of the shape functions do not effect the conformity of the truss dofs at the lattice sites. Therefore, it is possible to use linear elements with quadratic (or higher-order) elements in fully-resolved regions without having to worry about nonconformity. Fig. 5.6 gives an example of a higher-order mesh with the repUCs highlighted. Most of the mesh uses quadratic interpolation, but in the bottom left corner, linear elements are used to demonstrate how the choice of interpolation in the fully-resolved region is inconsequential.

5.2.2 Energy Approximation

Most of the summation rules in the literature are developed specifically for linear interpolation on a triangulated mesh, and do not naturally extend to higher-order interpolations. The exception to this are the quadrature summations rules that place a quadrature sampling unit cell at Gaussian quadrature points in all of the coarse-grained simplices, with weights given by

$$w = w_g \text{vol}(\Delta) \frac{\text{vol}(K)}{\text{vol}(\Omega_{UC})}, \quad (5.5)$$

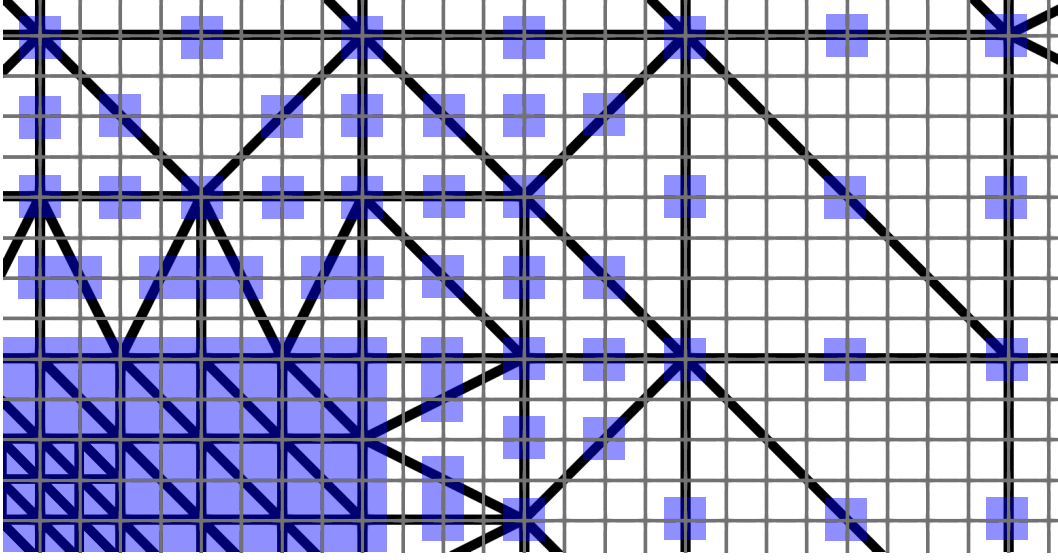


Figure 5.6: Example of the proposed higher-order interpolation strategy applied to a square lattice. The underlying truss lattice (grey) is approximated using the repUCs (blue) connected by the mesh (black).

where w_g is the Gaussian quadrature weight, $\text{vol}(\Delta) = 1/2$ or $1/6$ in 2D or 3D, respectively, is the volume of the unit simplex, and $\text{vol}(K)$ and $\text{vol}(\Omega_{UC})$ are the volumes of the coarse-grained element and unit cell, respectively. The first-order approximation Eq. (3.63) is used to generate the positions of the neighbor unit cell locations at the quadrature points, meaning quadrature sampling results in a local-nonlocal version of QC. The idea behind these quadrature rules is that, as the coarse-grained elements become large, the finite sum over the containing unit cells converges to an integral over the element, for which Gaussian quadrature is suitable.

However, as will be shown later, this summation rule can have considerable errors in moderately coarsened regions. Therefore, we propose a new summation rule based on the second-order summation rule described in Section 3.4.2 that retains nonlocal sampling unit cells in the coarse-grained regions. In two dimensions, the proposed summation rule for quadratic interpolation is analogous to the previously outlined second-order summation rule. Each vertex repUC is also a sample unit cell with a weight of 1. The edge sample unit cells (which now coincide with the mid-edge repUCs in the quadratic interpolation) have weights equal to the number of unit cells that lie on the edge. And lastly, there is a local sample unit cell at the barycenter of each coarse-grained element with a weight equal to the number of unit cells that lie on the interior

of the triangle (see Appendix B for details on how the weights are computed).

On the other hand, the second-order summation rule in three dimensions does not naturally extend to quadratic interpolation as the face sample unit cells lie on the faces of the tetrahedron rather than the edges, where the new repUCs are placed. So, we propose a summation rule similar to the second-order optimal summation rule where edge sample unit cells are used instead of face sample unit cells. Therefore, the proposed three-dimensional summation rule is one where each vertex repUC is a sample unit cell with a weight of 1, each mid-edge repUC is a sample unit cell with a weight equal to the number of unit cells on the edge (see Appendix B for details on how this is computed), and there is a local sample unit cell at the barycenter of each macroscopic element with the weight

$$w_{s,\text{inner}} = \frac{\text{vol}(K)}{\text{vol}(\Omega_{\text{UC}})} - \sum_{i=1} \omega_K(\mathbf{X}_i^{\text{rep}}) - \sum_{i=1} w_{i,\text{edge}} \omega_K(\mathbf{X}_i^{\text{edge}}), \quad (5.6)$$

where $\omega_K(\mathbf{X}_i^{\text{edge}})$ is the solid angle of the edge sample unit cell.

5.3 Performance of Higher-Order Methods

5.3.1 Elastic Constants Revisited

In order to test the combination of the quadratic interpolation and higher-order summations rules, we perform the exact same elastic constants tests of Section 4.1 with the newly introduced kinematics and energy approximation. For comparison purposes, the elastic constants tests are performed using both the quadrature summation rule and the slightly modified optimal summation rule described in Section 5.2.2. The errors in the predicted elastic constants are shown in Fig. 5.7 and Fig. 5.8.

As can be seen from the figures, the errors associated with the higher-order QC formulation are larger than errors seen with affine interpolation. The quadrature summation rule in two dimensions exhibits significant errors for all lattice topologies near the fully-resolved limit. However, the second-order summation rule is considerably more accurate (although less accurate than when used with linear interpolation) near full-resolution, and correctly matches the homogenized elastic moduli in both the fully-resolved and extremely coarse limits. In three dimensions, the accuracy of the quadrature and modified optimal summation rules are similar. Both summation rules have a larger

2D Lattice Moduli Errors

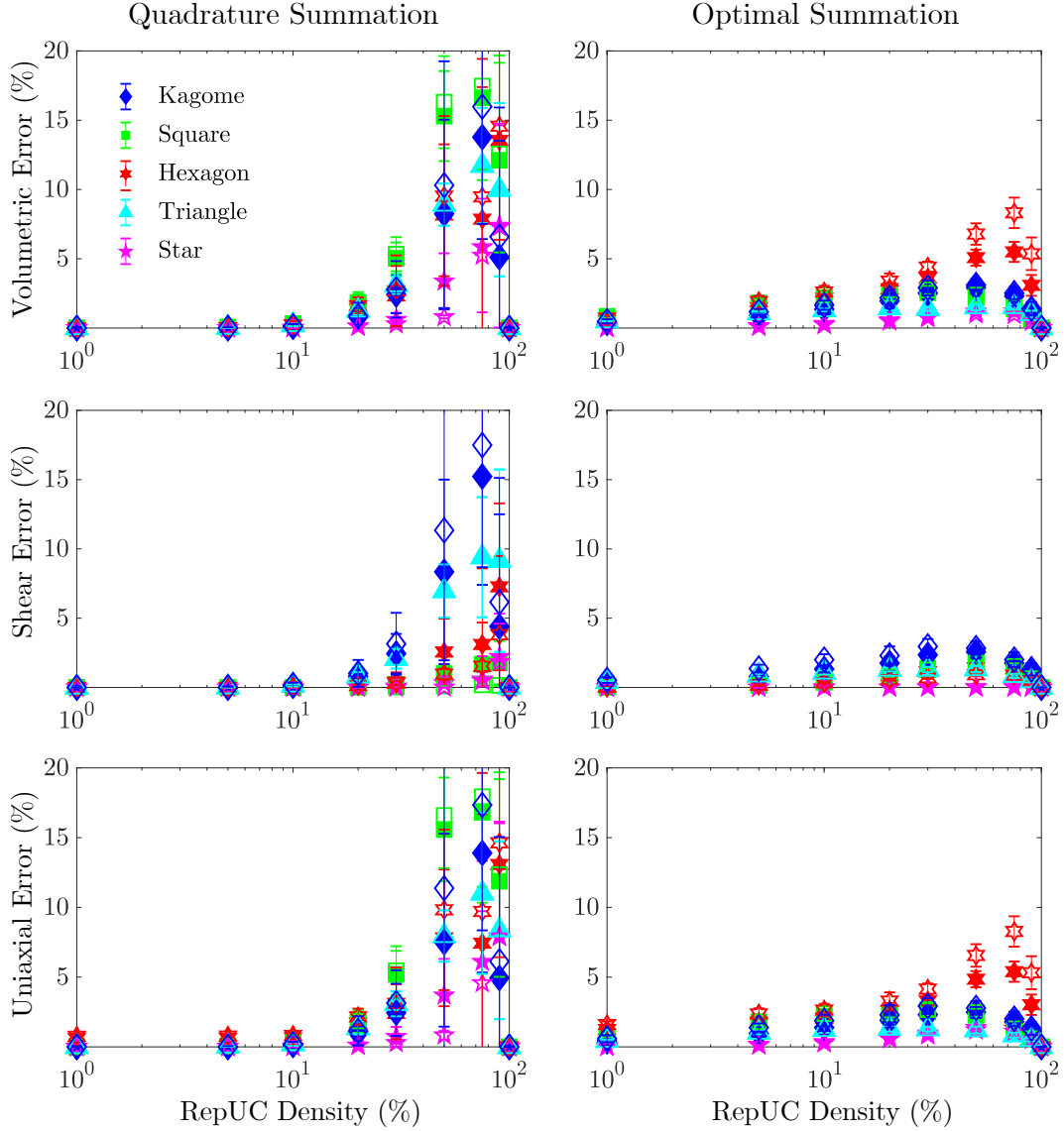


Figure 5.7: Errors in the linear elastic bulk, shear, and uniaxial moduli of 2D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with quadratic interpolation paired with quadrature (*left*) and optimal (*right*) summation rules and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.

3D Lattice Moduli Errors

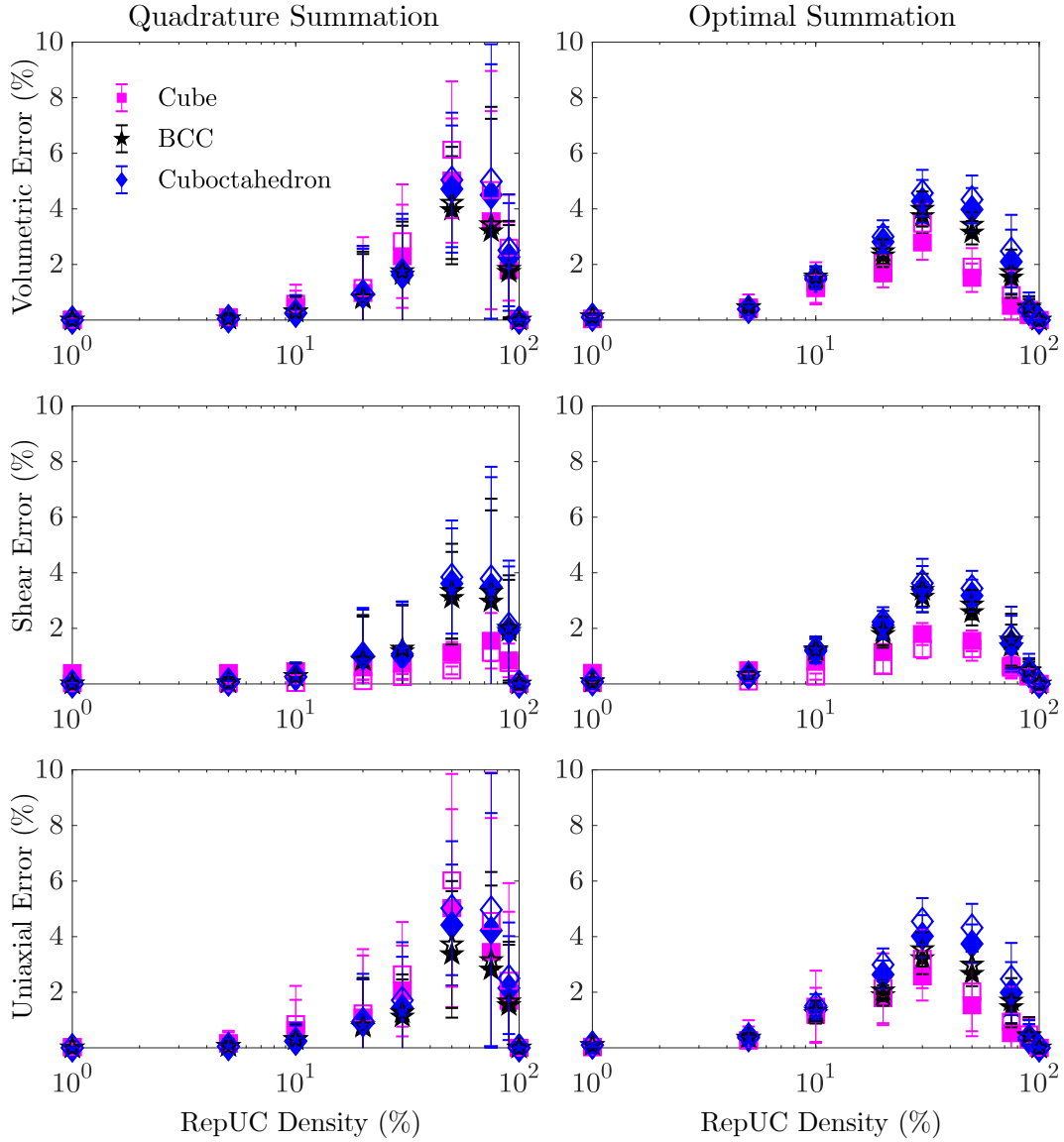


Figure 5.8: Errors in the linear elastic bulk, shear, and uniaxial moduli of 3D lattices at 20% (solid markers) and 1% (hollow markers) relative density, computed by coarse-grained truss QC simulations with quadratic interpolation paired with quadrature (*left*) and optimal (*right*) summation rules and random repUC selection for each level of repUC density. Markers correspond to the average error from 20 randomly coarsened meshes, and the bars represent the standard deviations of the errors.

error than in two dimensions, but the modified optimal summation rule is slightly more accurate near full resolution.

As in the linear interpolation case, all of the errors can be attributed purely to the error introduced in the summation rule. When the elastic constant simulations were performed using an exact summation rule, or when the meshes were uniformly coarsened, all of the errors were negligible. Therefore, the errors seen in Fig. 5.7 and Fig. 5.8 represent worse case scenarios, and much lower errors are expected in practice.

5.3.2 Cooks Membrane Revisited

To see how the higher-order interpolation and summation rules outlined in Section 5.2 perform, we revisit Cook's membrane simulation described in Section 5.1, but using the higher-order method. We expect that with the quadratic interpolation, the underlying lattice will be less constrained, and thus bending-dominated multilattices will be able to undergo inhomogeneous deformations without exhibiting stretch locking.

Since the Bravais lattice and stretching-dominated unit cells did not show any stretch locking in the first-order Cook's membrane simulations, here we focus on the bending-dominated multilattices: the star-shaped and hexagon lattices. To exaggerate stretch locking in the problem, the simulations were performed with each lattice having a low relative density of $\bar{\rho} = 1\%$. The quadratic meshes were created in a similar manner to the first-order meshes in Section 5.1, but using the method of first choosing "even" repUCs for the vertices, then adding the mid-edge repUCs afterwards as described in Section 5.2.1. For direct comparison, the linear and quadratic interpolation simulations were performed using the exact same repUCs; only the connectivity and order of interpolation is different. The simulations were performed using both the second-order summation rule, and an exact summation rule where every unit cell is used as a sample unit cell. This is done to show that any differences seen here are purely due to the quadratic interpolation rather than errors in energy approximation.

Figure 5.9 shows the strain energy of the Cook's membrane simulations of the star-shaped and hexagon lattices for various repUC densities using the quadratic interpolation. The results using linear interpolation from Section 5.1 are also copied here for comparison purposes.

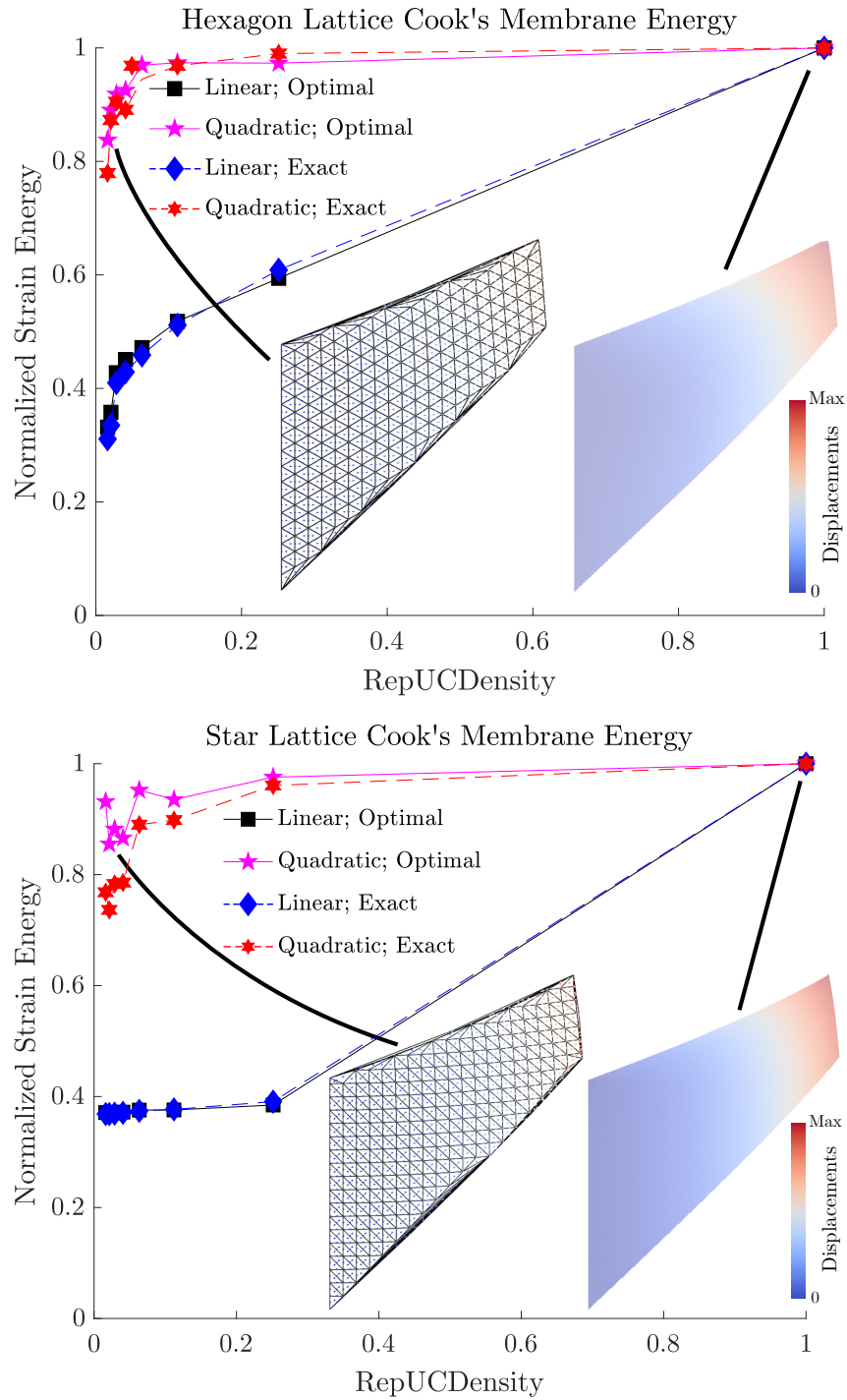


Figure 5.9: Normalized strain energy of the Cook's membrane simulations of the hexagon (*top*) and star-shaped (*bottom*) lattices with relative density of $\bar{\rho} = 1\%$. The insets show the deformed membranes and are colored by the magnitude of displacements (normalized by the maximum displacement of the fully-resolved simulation).

The most obvious and notable aspect of Fig. 5.9 is that neither the hexagon nor star-shaped lattices exhibit stretch locking when using the quadratic interpolation. The strain energies in both lattices closely match the strain energy of the fully-resolved simulations until the meshes become exceedingly coarse. Furthermore, it is clear that the improved performance is not due to the energy approximation (as is the case with reduced integration in finite elements), but is directly tied to the kinematics.

A surprising aspect of Fig. 5.9 is that the the hexagon lattice — a lattice that results in a nearly incompressible effective material — does not show any locking. If a hexagon lattice Cook’s membrane problem was simulated using FE^2 or homogenization techniques using fully-integrated Lagrange elements, the membrane would exhibit volumetric locking. More sophisticated techniques such as reduced integration, B-bar methods, or other multi-field variational methods (e.g. Hellinger-Reissner or Hu-Washizu principles) would need to be employed to accurately model the membrane. However, here we show that, when all of the dofs of the multilattice are kept at the macroscale, it is possible to accurately model nearly incompressible effective materials without using multiple fields or reduced integration.

5.3.3 Boundary Layer Revisited

In Section 4.2, it was seen that the adaptive QC versions of the boundary layer method for predicting the brittle fracture toughness of 2D lattices was accurate for the triangle and kagome lattices but was inaccurate for the hexagon and star-shaped lattices due to stretch locking. Here, we revisit the adaptive boundary layer method simulations for the hexagon and star-shaped lattice using the quadratic interpolation and higher-order summation rule described in Section 5.2.

The geometry and boundary conditions are almost identical to the those in Section 4.2, with the small difference being the notch width. Since the geometry is created by first creating vertex repUCs that lie on sites with even Bravais coordinates, the resolution in creating the geometry is two unit cells. Therefore, the notch in the lattices here have a width of two unit cells rather

than notches with a width of one unit cell as in Section 4.2⁵. The simulations are performed with a slender truss with relative density $\bar{\rho} = 1\%$ in order to emphasize any stretch locking that would occur.

The refinement algorithm used here with quadratic interpolation is analogous to the refinement algorithm outlined in Section 3.5 and used in the affine interpolation simulations in Section 4.2. One difference is that the refinement algorithm searches for new *vertex* repUCs, so that the new repUC locations are chosen to be at the nearest "even" (in Bravais coordinates) unit cell locations to the midpoint of the longest element edges. The other modification is that we use the average deformation gradient across the quadratic simplex when computing the refinement metric. We use the average deformation gradient since it is not constant in each macroscopic element as it is with affine interpolation.

Since relatively large errors were seen in the elastic constants section, the simulations here are performed with quadratic interpolation using both the higher-order summation rule described in Section 5.2, but also with an exact summation rule where the energy is computed with a sum over all beams in the simulation. Figure 5.10 shows the error in strain energy and in the maximum stress at the notch tip for the hexagon and star-shaped lattices as the quadratic mesh is refined using the second invariant refinement criterion. Clearly, the quadratic interpolation performs significantly better at approximating the true solution than the linear interpolation in Section 4.2. The errors in energy and maximum tensile stress near the notch tip are significantly lower even with the extremely coarse original mesh, showing that the stretch locking is mitigated.

Even though stretch locking is not an issue when using quadratic interpolation, there is some undesirable behavior in the convergence plots. The most obvious issue is that, while the strain energy of the star lattice converges to the fully-resolved case, the maximum tensile stress near the notch tip still has a large error at repUC densities greater than 10%. This is attributed to the naïve criterion being used for the mesh refinement. Possibly using goal-oriented refinement technique as in [68] could be used to improve the convergence of the maximum stress, but it is outside the scope of this thesis.

⁵The same adaptive QC simulations using linear interpolation were performed on the hexagon and star-shaped lattices with this larger notch and showed similar behavior to those with the smaller notch. Any improvement seen in the section is therefore not due to the slight change in geometry, and can be attributed to the change in the kinematic constraints.

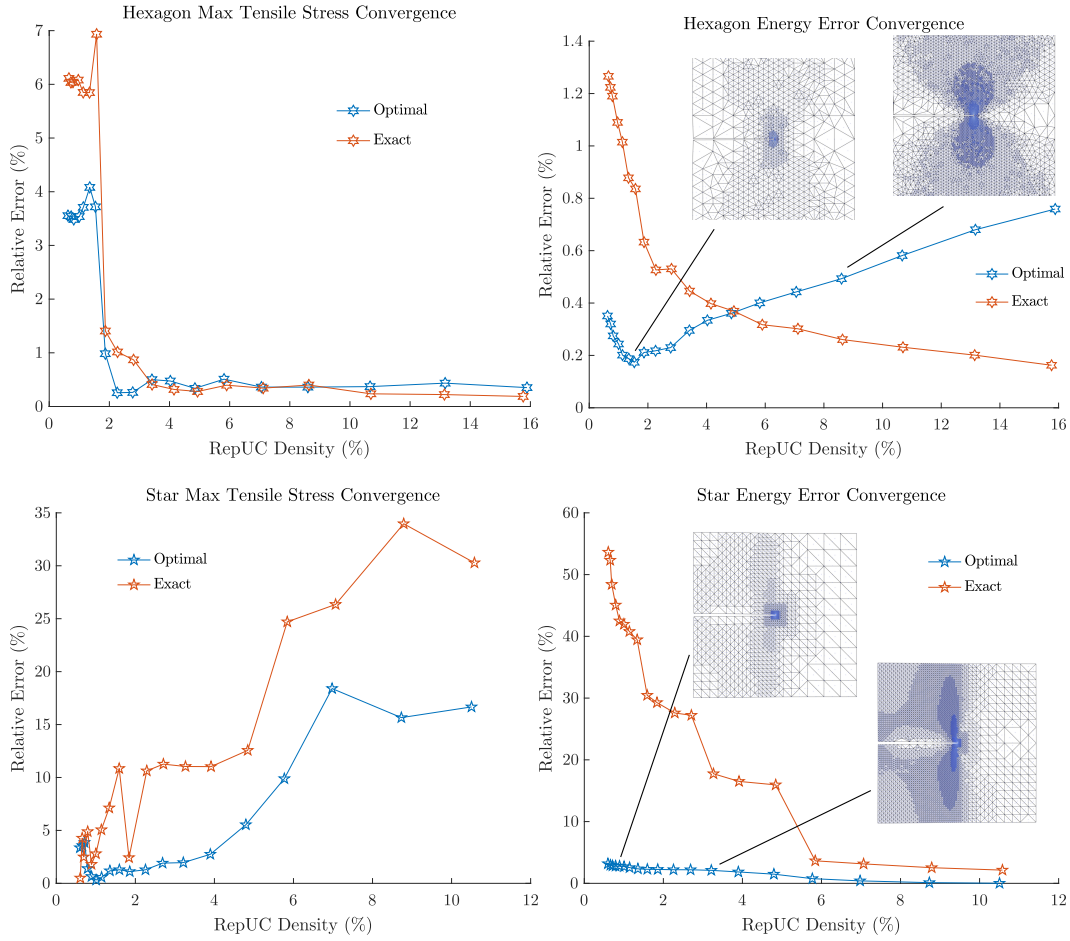


Figure 5.10: Convergence of the strain energy and maximum tensile stress in a strut near the notch tip for the hexagon and star-shaped lattice coarse-grained boundary layer simulations using the quadratic interpolation and the optimal (blue) and exact (orange) summation rules.

The other unusual behavior in the convergence plots is the fact that the error in strain energy for the hexagon lattice increases as the mesh is refined when the optimal summation rule is used. This error is due to the fact that the summation rule (as shown in Section 5.3.1) accurately approximates the energy of the lattice in extremely coarse regions and in fully-resolved regions, but is not as accurate in moderately coarsened regions. Therefore, as the mesh is refined and more of the domain enters this moderately coarsened region, the kinematics become more accurate, but the error due to the energy approximation increases.

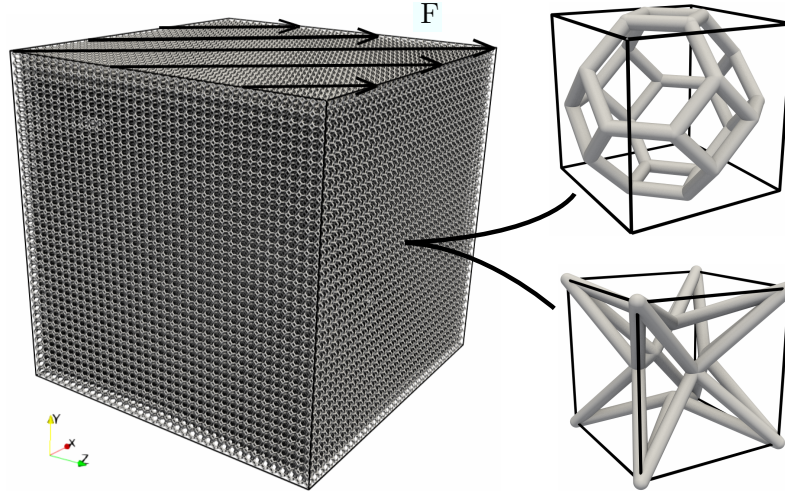


Figure 5.11: A lattice block with distributed shear load placed on the top. The cube is made by tessellating either the tetrakaidecahedron (*top right*) or the 3D star-shaped (*bottom right*) unit cell.

5.3.4 Lattice Block With Distributed Shear Load

In the previous sections, the higher-order QC method was shown to mitigate stretch locking for two-dimensional test problems. Here, we test the formulation for three dimensional bending-dominated multilattices. Specifically, we perform shear tests on the tetrakaidecahedron lattice and a three-dimensional version of the 2D star-shaped lattice (see Fig. 5.11). The tetrakaidecahedron is a bending-dominated multilattice that has a stretching-dominated bulk modulus, and thus behaves like an incompressible effective material. On the other hand, all of the macroscopic deformation modes of 3D star-shaped lattice are bending-dominated (including volumetric deformation).

For both lattice topologies, a unit cell with side length equal to 1 (each truss member has length $L = \sqrt{2}/4$ for the tetrakaidecahedron and $L = 3/4$ for the 3D star-shaped lattice) is tessellated in a $36 \times 36 \times 36$ cube to create the geometry shown in Fig. 5.11. A constant distributed load is placed over the top of the lattice block (i.e. applied to all nodes in the unit cells on the top surface of the block) in the $(1, 0, 1)$ direction, as shown in Fig. 5.11. In order to exaggerate stretch locking, both lattices are taken to have a relative density of $\bar{\rho} = 1\%$. The translational and rotational dofs of the bottom of the lattice are fixed in place.

As with the Cook’s membrane simulation, the block is coarse-grained in a uniform fashion using both linear and quadratic interpolation paired with both

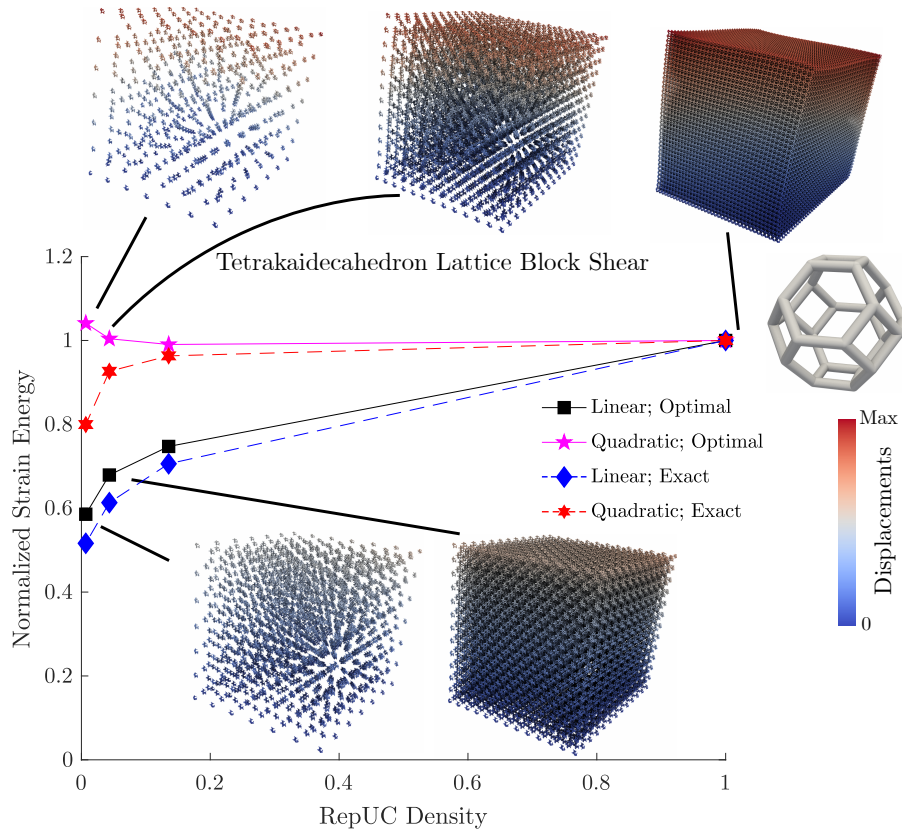


Figure 5.12: Strain energy in the coarse-grained QC simulations of the tetrakaidecahedron lattice block undergoing a distributed shear load across the top surface. The insets show the deformed configurations of all of the sampling unit cells, and are color-coded based on the displacements.

the optimal summation rules and exact summation rules for comparison with the fully-resolved simulation. The total strain energy in the lattice is computed for each simulation to be used as a metric for stretch locking. Figure 5.12 and Fig. 5.13 show the strain energy in the lattices as computed by the linear and quadratic versions of the QC method at various repUC densities.

Similar to the 2D Cook’s membrane simulations, the response of the 3D bending-dominated multilattices are overly stiff when using linear interpolation. The tetrakaidecahedron lattice block behaves almost twice as stiff as the fully-resolved simulation when using every 6th unit cell as a repUC in each direction, while the coarse-grained 3D star-shaped lattice with linear interpolation is almost 10 times as stiff. However, when quadratic interpolation is used with an exact summation rule, the tetrakaidecahedron and 3D star-shaped lattices had strain energies above 0.8 times the fully-resolved simulation, which suggests that stretch locking does not occur with quadratic interpolation.

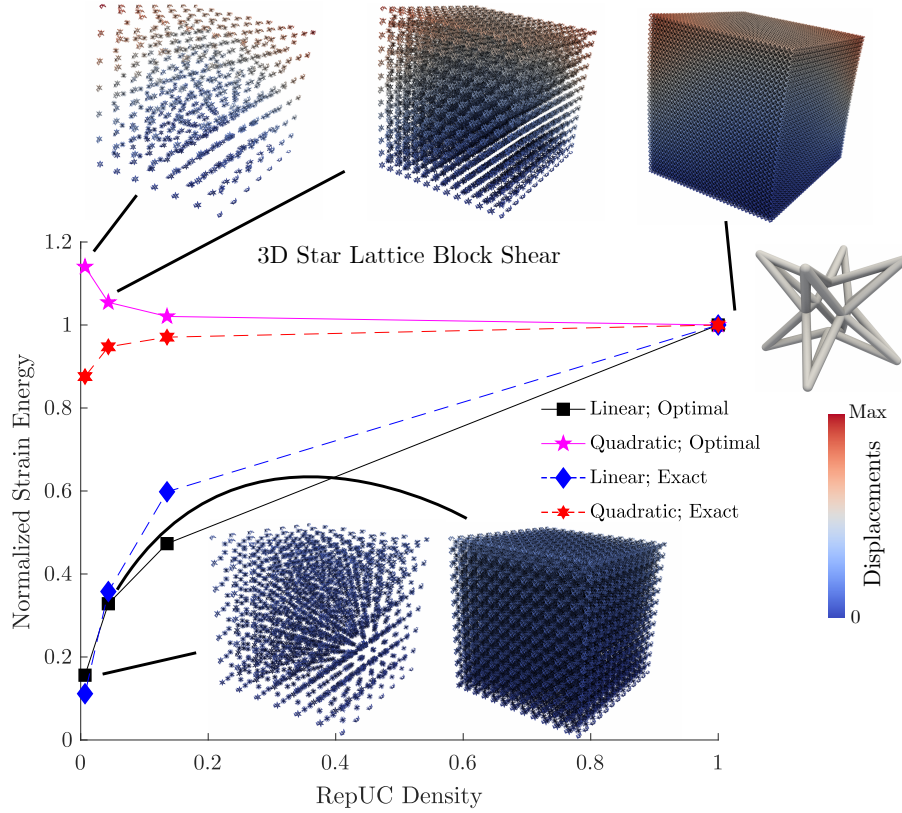


Figure 5.13: Strain energy in the coarse-grained QC simulations of the 3D star-shaped lattice block undergoing a distributed shear load across the top surface. The insets show the deformed configurations of all of the sampling unit cells, and are color-coded based on the displacements.

For both the tetrakaidecahedron and 3D star-shaped lattices, the optimal summation rule developed in Section 5.2.2 begins to disagree with the fully-resolved sampling at low repUC densities. This is because the $36 \times 36 \times 36$ unit cell geometry has a significant number of unit cells that lie on the boundary of the mesh ($\approx 1/6$ of the unit cells lie on the boundary), which are not properly accounted for with the summation rule. In the second-order optimal summation rule for linear interpolation, face sampling unit cells were chosen with weights to account for unit cells on the boundaries. However, the summation rule in Section 5.2.2 moved the face sampling unit cells to the element edges and they only account for unit cells that lie on element edges, not faces. Therefore, the summation rule does a poor job at approximating the number of unit cells in geometries with a significant number of boundary unit cells when compared to inner unit cells. The under-weighting of the summation rule results in a more compliant behavior, which is seen by the larger strain energies at lower repUC

densities. Improved summation rules that better approximate the boundary unit cells could be used to minimize this error, but it is outside the scope of this thesis.

As with the hexagon lattice in the Cook’s membrane simulation, the quadratically-interpolated coarse-grained simulations of the nearly-incompressible tetrakaidecahedron lattice matched the fully-resolved simulation well even for very coarse meshes. If this lattice was to be modeled using numerical homogenization techniques (e.g. the FE^2 method), more sophisticated techniques to suppress volumetric locking (e.g. reduced integration, B-bar methods) would be needed to accurately model the lattice. Here, we retain accuracy without resorting to using these techniques.

NUMERICAL CONSIDERATIONS

6.1 Structure of a Simulation

The QC method presented in this thesis was implemented in an in-house object-oriented finite element code. In this chapter, we give an overview of the implementation, and the computational costs are discussed. We include the flow chart Fig. 6.1 to give the reader a high level outline of the structure of a simulation.

6.1.1 Setting up the Problem

After choosing the lattice topology, structural element to model each truss member, as well as the underlying material model, repUCs must be chosen and mesh created. The periodic vectors of the lattice topology are used as basis vectors for potential repUC locations, i.e. the unit cell locations are chosen from points in the set in Eq. (2.2) with $\{\mathbf{a}_1, \dots, \mathbf{a}_d\}$ being the periodic vectors of the lattice topology.

For the simulations presented in this thesis, the repUC locations are either chosen such that there is a roughly uniform distribution of repUCs across the domain (e.g. Fig. 5.1), or such that there is a small fully-resolved region, and the mesh gradually coarsens away from this region (e.g. Fig. 4.11). If quadratic interpolation is used, only the vertex repUCs are chosen at this stage.

After the repUC locations are chosen, the meshing library CGAL [51] is used to create the element connectivity of a simplicial mesh of the convex hull of the repUCs. If the desired geometry is nonconvex, then various techniques (including adding additional points and manually deleting some elements) outlined in Amelang [2] are used to create the nonconvex geometry.

If the quadratic interpolation formulation of Section 5.2 is used, then additional repUCs are placed at the midpoint of each simplicial element edge. Finally, the mesh consisting of all of the repUCs, the element connectivity, and the shape functions is created. This mesh is used throughout the QC simulation to determine the kinematics (e.g. nonzero shape function values) for unit cells. This mesh is identical to a typical finite element mesh, except the nodes of the

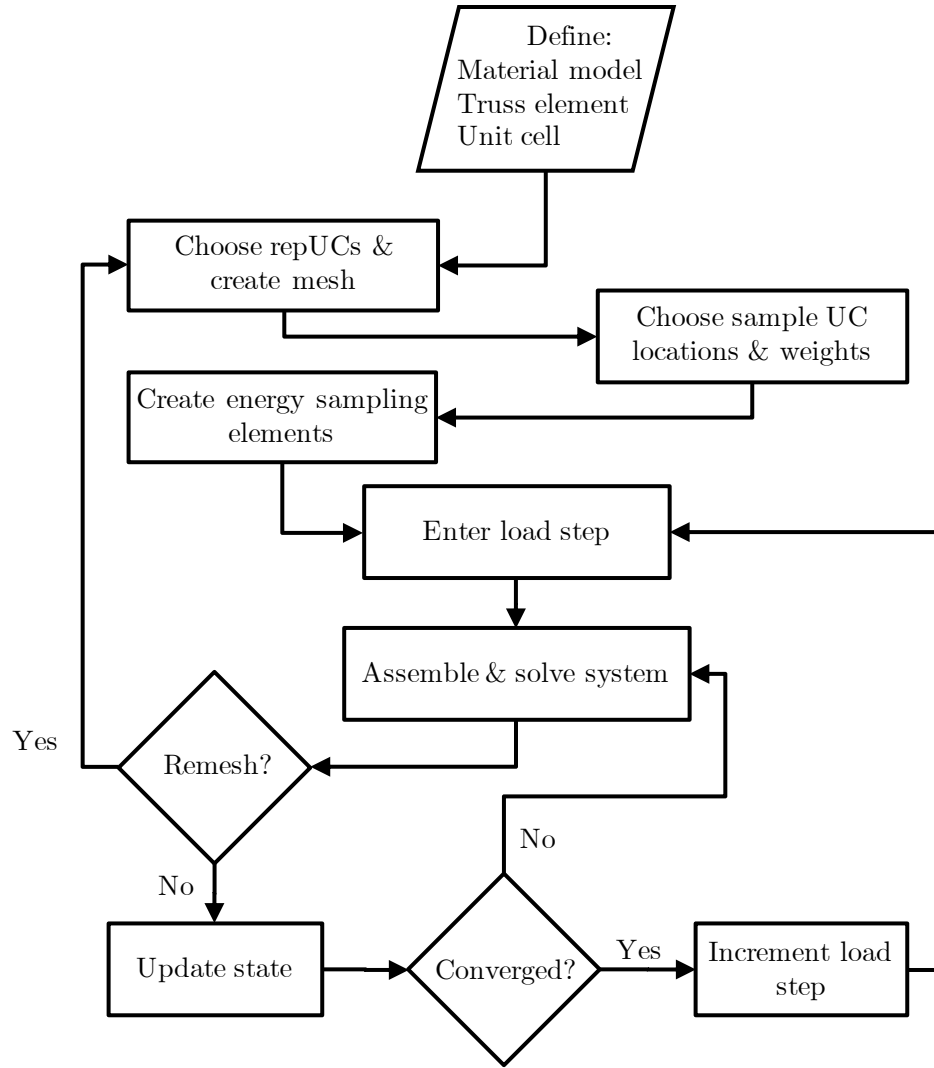


Figure 6.1: Program flow chart.

mesh are the repUC locations, and the dofs at each mesh node consist of all of the dofs of the repUC.

6.1.2 Populating Sampling Unit Cells

After all of the kinematics are set up, the locations and weights of the sampling unit cells are computed for energy calculation. Unlike the finite element method where Gaussian quadrature and isoparametric mapping can be used to determine the weights and locations of the quadrature points a-priori, all of the weights and locations of the sampling unit cells in the QC method must be computed after the kinematics are defined in order to set up the simulation.

Regardless of the summation rule being used, the corresponding locations and

weights of the sampling unit cells are computed. After all of the sampling unit cell locations are chosen, the corresponding structural elements that are used to model the truss members in each sampling unit cell must be created. These *sampling truss members* store the following properties:

- sampling weight, w_t
- the lattice index of each node (i.e. which lattice in the multilattice each node is connected to). As in Section 3.1, the superscript j is used to denote the lattice index (e.g. $\boldsymbol{\varphi}^j(\mathbf{X}_i)$ are the dofs of the j^{th} node of the multilattice at the unit cell location \mathbf{X}_i).
- nonzero shape functions at each node $N_r(\mathbf{X}_i)$. Both the global indices of the nonzero shape functions, denoted by the set

$$\text{supp}(\mathbf{X}_j) = \{r \mid N_r(\mathbf{X}_j) \neq 0\} \quad (6.1)$$

and the shape function values themselves $N_r(\mathbf{X}_i)$, $r \in \text{supp}(\mathbf{X}_1) \cup \text{supp}(\mathbf{X}_2)$ are stored. This is such that the dofs of the endpoints of the truss member can be efficiently computed from

$$\boldsymbol{\varphi}^j(\mathbf{X}_i) = \sum_{r \in \text{supp}(\mathbf{X}_i)} N_r(\mathbf{X}_i) \boldsymbol{\varphi}_r^j, \quad (6.2)$$

where $\boldsymbol{\varphi}_r^j$ are the dofs of the r^{th} repUC.

For truss members that connect nodes within the same unit cell (i.e. the set \mathcal{E}_u defined in Section 3.1), the weight is equal to the unit cell sampling weight ($w_t = w_s$), and the two nodes of the truss member have the same non-zero shape functions and shape function values. For truss members that are connected to nodes in neighboring unit cells (i.e. the set \mathcal{E}_u^n), the sampling weight of the truss element is half of the sample unit cell weight $w_t = w_s/2$ (to avoid double counting) and the nonzero shape functions depend on whether the sampling unit cell is a local sampling unit cell or not. If it is a local sampling unit cell, then the dofs of the neighboring unit cell are defined by the first-order Taylor expansion of the dofs at the sampling unit cell location as in Eq. (3.63), and therefore both nodes of the truss share the same nonzero shape functions (i.e. $\text{supp}(\mathbf{X}_1) = \text{supp}(\mathbf{X}_2)$). Writing the dofs of the nodes of

the truss element in terms of their nonzero shape functions and repUC dofs, one obtains

$$\varphi_{\text{loc}}^j(\mathbf{X}_i) = \sum_{r \in \text{supp}(\mathbf{X}_s)} \left(N_r(\mathbf{X}_s) + \frac{\partial N_r(\mathbf{X})}{\partial \mathbf{X}} \Big|_{\mathbf{X}=\mathbf{X}_s} (\mathbf{X}_i - \mathbf{X}_s) \right) \varphi_r^j, \quad (6.3)$$

where \mathbf{X}_s and \mathbf{X}_i are the positions of the sampling unit cell and its neighbor, respectively. Therefore, nodes of the truss members in local sampling unit cells are taken to have the fictitious shape function values

$$\tilde{N}_r(\mathbf{X}_i) = N_r(\mathbf{X}_s) + \frac{\partial N_r(\mathbf{X})}{\partial \mathbf{X}} \Big|_{\mathbf{X}=\mathbf{X}_s} (\mathbf{X}_i - \mathbf{X}_s). \quad (6.4)$$

If the sampling unit cell is not a local sampling unit cell and the fully-nonlocal energy is computed, then the two nodes of the truss member could potentially have different nonzero shape functions (e.g. if the truss element crosses boundaries of the macroscopic mesh as in Fig. 6.2). It is also possible that the potential adjacent unit cell could not be in the simulation (e.g. if the sampling unit cell is on the boundary of the domain), in which case the truss elements that would be connected to that neighboring unit cell are ignored.

The fact that the two nodes of the truss member could lie in different mesh elements is a fundamental difference between the nonlocal QC method and finite element method used for solving partial differential equations. In the finite element method, the local energy is usually sampled at quadrature points, and can be computed given the dofs of the nodes of the element that contains the quadrature point. However, the repUC dofs needed in order to compute the energy of a sampling truss element in the nonlocal QC method are not defined by a the repUCs of a macroscopic element (see, e.g. the green truss members in Fig. 6.2). However, it is possible to lump sampling truss members into sets with the same nonzero shape function indices (i.e. with identical support $\text{supp}(\mathbf{X}_1) \cup \text{supp}(\mathbf{X}_2)$, see Fig. 6.2), such that the energy of the system can be expressed as a sum over these so-called *energy sampling elements*:

$$\mathcal{I}^h(\tilde{\varphi}) = \sum_{e \in \mathcal{E}_s} \sum_{t \in e} w_t W_t \left(\sum_{r \in \text{supp}(\mathbf{X}_1)} N_r(\mathbf{X}_1) \varphi_r^j, \sum_{r \in \text{supp}(\mathbf{X}_2)} N_r(\mathbf{X}_2) \varphi_r^j \right), \quad (6.5)$$

where \mathcal{E}_s is the set of energy sampling elements, and \mathbf{X}_1 and \mathbf{X}_2 are the locations of the unit cells that contain the two nodes of the truss member.

In Section 3.1, we expressed the energy of the system as a sum over unit cells to draw the comparison to atomistics and develop the QC framework. However,

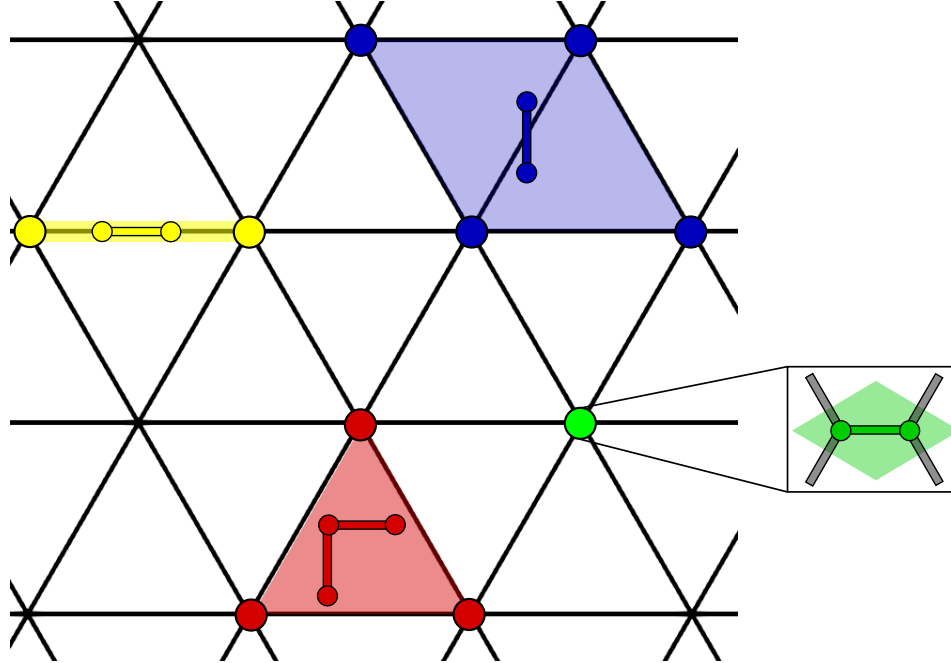


Figure 6.2: Examples of sampling truss elements (with small nodes) and their corresponding energy sampling elements (with larger nodes). The colors of the truss members match the nodes of the mesh needed to compute the energy of the truss member (i.e. the repUCs with nonzero shape functions at the truss member nodes). The number of nonzero shape functions can vary - examples of truss members with 1 (green), 2 (yellow), 3 (red), and 4 (blue) repUC dependencies are shown.

it is more convenient to express the energy as in Eq. (6.5) for implementation in a finite element code. By organizing things in this manner, these energy sampling elements act just like traditional finite elements, except the energy sampling elements form an overlapping mesh (distinct from the mesh used for kinematics), and the truss members are used in lieu of quadrature points.

6.1.3 Assembling the Global Quantities

Although there are algorithms that can be used to minimize an energy potential using only the energy itself, it is desirable to use information coming from first- and second-order derivatives of the energy with respect to the global dofs (i.e. the force and stiffness matrices respectively) to minimize the energy in a more efficient manner (see Section 6.1.5 for more information on solvers being used). Here, we give algorithms used to assemble the global energy (Algorithm 2), force (Algorithm 3), and stiffness (Algorithm 4) of the simulation (without applying boundary conditions, which will be discussed in

Section 6.1.4).

In the algorithms, we use subscripts for global quantities (e.g. \mathbf{F}_r , \mathbf{K}_{r_1, r_2}) to refer to the sub-vector or sub-matrix of the global vector or stiffness matrix corresponding to the repUC indices r . Similarly, we use superscripts (e.g. \mathbf{F}^p , \mathbf{K}^{p_1, p_2}) to refer to the sub-vectors or sub-matrices related to the dofs of the p^{th} Bravais lattice in the multilattice.

Algorithm 2 Compute Global Energy

```

 $W = 0;$ 
for each energy sampling element  $E_s$  do
  for each truss member  $t$  in  $E_s$  do
    for each point  $p$  of  $t$  do
      Get dofs at point,  $\boldsymbol{\varphi}^j(\mathbf{X}_p) = \sum_{r \in \text{supp}(\mathbf{X}_p)} N_r(\mathbf{X}_p) \boldsymbol{\varphi}_r^j$ .
      Calculate truss energy  $W_t(\boldsymbol{\varphi}^j(\mathbf{X}_1), \boldsymbol{\varphi}^j(\mathbf{X}_2))$ 
       $W \leftarrow W + w_t W_t$ 
    Add energy contribution from external forces

```

Algorithm 3 Compute Global Force Vector

```

 $\mathbf{F} = \mathbf{0}$ 
for each energy sampling element  $E_s$  do
  for each sampling truss member  $t$  in  $E_s$  do
    for each point  $p$  of  $t$  do
      Get dofs at point,  $\boldsymbol{\varphi}^j(\mathbf{X}_p) = \sum_{r \in \text{supp}(\mathbf{X}_p)} N_r(\mathbf{X}_p) \boldsymbol{\varphi}_r^j$ .
      Calculate truss force  $\mathbf{F}_t(\boldsymbol{\varphi}^j(\mathbf{X}_1), \boldsymbol{\varphi}^j(\mathbf{X}_2))$ 
    for each node  $p$  of  $t$  do
      for each repUC,  $r \in \text{supp}(p)$  do
         $\mathbf{F}_r^p \leftarrow \mathbf{F}_r^p + w_t N_r(\mathbf{X}_p) \mathbf{F}_t$ 
    Add force contribution from external forces

```

6.1.4 Treatment of Boundary Conditions

Essential boundary conditions can be treated numerically in two ways. That is, the boundary conditions $\tilde{\boldsymbol{\varphi}}_d = \tilde{\boldsymbol{\varphi}}_0$ can be treated as strict constraints, or the constraint can be included into the optimization problem using Lagrange multipliers. The former will be referred to as *strong* enforcement of the boundary conditions and is equivalent to the problem

$$\begin{aligned} \min_{\text{s.t. } \tilde{\boldsymbol{\varphi}}_d = \tilde{\boldsymbol{\varphi}}_0} \{ \mathcal{I}(\tilde{\boldsymbol{\varphi}}_i) \}, \end{aligned} \quad (6.6)$$

Algorithm 4 Compute Global Tangent Matrix

```

K = 0
for each energy sampling element  $E_s$  do
  for each sampling truss member  $t$  in  $E_s$  do
    for each point  $p$  of  $t$  do
      Get dofs at point,  $\varphi^j(\mathbf{X}_p) = \sum_{r \in \text{supp}(\mathbf{X}_p)} N_r(\mathbf{X}_p) \varphi_r^j$ .
      Calculate stiffness matrix,  $\mathbf{K}_t(\varphi^j(\mathbf{X}_1), \varphi^j(\mathbf{X}_2))$ 
    for each node  $p_1$  of  $t$  do
      for each node  $p_2$  of  $t$  do
        for each repUC  $r_1 \in \text{supp}(p_1)$  do
          for each repUC  $r_2 \in \text{supp}(p_2)$  do
             $\mathbf{K}_{r_1, r_2}^{p_1, p_2} \leftarrow \mathbf{K}_{r_1, r_2}^{p_1, p_2} + w_t N_{r_1}(\mathbf{X}_{p_1}) N_{r_2}(\mathbf{X}_{p_2}) \mathbf{K}_t^{p_1, p_2}$ 
      Add stiffness contribution from external forces
  
```

while the latter is referred to as *weak* enforcement of the boundary conditions and is associated with the problem

$$\min_{\tilde{\varphi}} \max_{\lambda} \{ \mathcal{I}(\tilde{\varphi}) + \sum_d \lambda_d^T (\tilde{\varphi}_d - \tilde{\varphi}_0) \},^1 \quad (6.7)$$

where λ_d are the Lagrange multipliers. We note that Eq. (6.6) and Eq. (6.7) have the same solutions, but have differences when it comes to numerical implementation. For the remainder of this section, we will use the notation that the set of dofs are partition based on whether they have a (d)irichlet boundary condition or they are (f)ree (i.e. $\tilde{\varphi}^T = \{\tilde{\varphi}_d^T, \tilde{\varphi}_f^T\}$).

Strong Enforcement of Boundary Conditions

When treating the essential boundary conditions as constraints, then a trial set of displacements given to the solver (e.g. an initial guess in an iterative solver) should exactly satisfy the boundary conditions. If that is not the case, the solver would need to manually change the trial displacement field to satisfy the boundary conditions². When using a force-based solve (e.g. steepest descent) this can be achieved by changing the solution vector $\tilde{\varphi}$ and force vector \mathbf{F} with

$$\tilde{\varphi} = \begin{bmatrix} \tilde{\varphi}_0 \\ \tilde{\varphi}_f \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}_f \end{bmatrix}. \quad (6.8)$$

¹A constant could also be included in the lagrange multiplier term such that it becomes $\sum_d C_d \lambda_d (\tilde{\varphi}_d - \tilde{\varphi}_0)$, which may be tuned such that the system is well-conditioned. This is left out of the analysis for simplicity, but all of the analysis that follows does not depend on $C_d = 1$.

²Or, the solver could throw an out-of-domain error.

Or if a higher-order solver such as Newton's method is used, the constraints can be enforced by zeroing out rows and columns in the stiffness matrix \mathbf{K} and editing the force vector as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{ff} \end{bmatrix} \begin{bmatrix} \Delta\tilde{\boldsymbol{\varphi}}_d \\ \Delta\tilde{\boldsymbol{\varphi}}_f \end{bmatrix} = - \begin{bmatrix} \mathbf{0} \\ \mathbf{F}_f \end{bmatrix}. \quad (6.9)$$

Of course, in this case it is not even necessary to assemble the stiffness matrix and forces for the dofs with dirichlet boundary conditions, as the linear system $\mathbf{K}_{ff}\Delta\tilde{\boldsymbol{\varphi}}_f = -\mathbf{F}_f$ can be solved to determine the update to the solution vector.

While this method has some desirable properties — namely that it retains the symmetry of the stiffness matrix and enforces the dirichlet boundary conditions exactly — it is not always desirable since it is difficult to supply the solver with good initial guesses that exactly satisfy the boundary conditions. And if the supplied initial guess does not exactly match the boundary conditions, the manually modified guess $\tilde{\boldsymbol{\varphi}}^T = \{\tilde{\boldsymbol{\varphi}}_0^T, \tilde{\boldsymbol{\varphi}}_f^T\}$ could be poor and lead to suboptimal convergence. For example, in a simulation with multiple load steps like the hole-in-plate problem of Section 4.5, using the solution from the previously converged load step as the initial guess for the current load step would result in a modified solution vector where *all* of the change in deformation between load steps is carried by the truss members at the edges of the plate. The 1% macroscopic strain difference between load steps would result in each end of the plate moving by an amount roughly equal to the length of a truss member, such that the axial strains in the bars near the boundaries would be roughly 100% in the modified solution vector. If the truss members are highly nonlinear (with geometric and/or material nonlinear), iterative solvers could have trouble converging from such a highly deformed state.

Weak Enforcement of Boundary Conditions

Weak enforcement of the boundary conditions have the advantages over the strong enforcement that it does not require a trial solution to exactly match boundary conditions. However, it appears that the system that needs to be solved is larger since the Lagrange multipliers are included as dofs, but we will show that this can be avoided in some circumstances. If an algorithm like

Newton's method is used, the linear system to be solved is

$$\begin{bmatrix} \mathbf{K}_{dd} & \mathbf{K}_{df} & \mathbf{I} \\ \mathbf{K}_{fd} & \mathbf{K}_{ff} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \tilde{\boldsymbol{\varphi}}_d \\ \Delta \tilde{\boldsymbol{\varphi}}_f \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_d + \boldsymbol{\lambda} \\ \mathbf{F}_f \\ \tilde{\boldsymbol{\varphi}}_d - \tilde{\boldsymbol{\varphi}}_0 \end{bmatrix}. \quad (6.10)$$

Notice that the Lagrange multipliers are not needed to be computed explicitly if only the dofs of the truss are desired, and the subsystem

$$\begin{bmatrix} \mathbf{K}_{fd} & \mathbf{K}_{ff} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \tilde{\boldsymbol{\varphi}}_d \\ \Delta \tilde{\boldsymbol{\varphi}}_f \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_f \\ \tilde{\boldsymbol{\varphi}}_d - \tilde{\boldsymbol{\varphi}}_0 \end{bmatrix} \quad (6.11)$$

can be solved instead. This system can be further reduced by analytically solving for $\Delta \tilde{\boldsymbol{\varphi}}_d$ and substituting back into the equation to obtain

$$\mathbf{K}_{ff} \Delta \tilde{\boldsymbol{\varphi}}_f = -\mathbf{F}_f + \mathbf{K}_{fd}(\tilde{\boldsymbol{\varphi}}_d - \tilde{\boldsymbol{\varphi}}_0). \quad (6.12)$$

Notice that this linear system that needs to be solved for the update in the solution field is the same size as the reduced system when using the strong enforcement of boundary conditions, but requires the additional (relatively small) cost of evaluating the extra term $\mathbf{K}_{fd}(\tilde{\boldsymbol{\varphi}}_d - \tilde{\boldsymbol{\varphi}}_0)$. However, the benefit from fact that the algorithm can handle any solution vector as an initial guess far outweighs the the drawback of having to compute the extra term. Using the example of the hole-in-plate simulation of Section 4.5, the solution vector from the previously converged step could be used as an initial guess for the current load step without highly deforming the truss members at the plate edges, leading to much better convergence of the nonlinear solver. For that reason, we use the weak enforcement of boundary conditions when minimizing the total potential energy of the system.

6.1.5 Solving the System

In order to find the minimizers of the QC-approximated energy (3.56), we leverage solvers and optimization techniques provided by PETSc/TAO [8–10]. The global energy, forces, and tangent stiffness computed using Algorithms 2 to 4, and modified by enforcing boundary conditions in a weak fashion (see above), are passed to PETSc/Tao to use their suite of preconditioners and (non)linear solvers. The workhorse of the PETSc/TAO package is Newton's method paired with either a direct linear solver (e.g. LU or Cholesky decomposition) or a preconditioned Krylov solver (e.g. algebraic-multigrid-preconditioned gmres

solver) and a line search. PETSc also comes with many other tools including matrix-free solvers, Quasi-Newton methods, and the ability to combine various linear and nonlinear solvers and preconditioners [26].

For reasonably sized linear problems, we found that using the reduced system Eq. (6.12) paired with the Cholesky decomposition solver provided by MUMPS [4, 5]³ offered a very robust and efficient solver.

However, for extremely large linear systems, like those seen in the 3D volumetric locking test in Section 5.3.4 in which the direct solver required too much memory, we found other preconditioners to be useful. In particular, the method of iteratively solving for displacements and rotations of the truss separately (called the *fieldsplit* preconditioner in PETSc) was used to solve the large linear systems⁴.

For stretching-dominated bravais lattices (e.g. triangle lattice), the algebraic multigrid preconditioner in PETSc proved to be a very efficient preconditioner for iterative linear solvers⁵. However, we were unsuccessful in getting the algebraic multigrid preconditioner to yield good convergence properties for bending-dominated lattices or multilattices (e.g. square, hexagon, kagome, tetrakaidecahedron, etc.). Algebraic multigrid uses heuristics that are known to yield sub-par convergence for ill-conditioned systems like those seen in anisotropic or nearly-incompressible elasticity, so it is not surprising that the black-box preconditioner is not well-suited for the ill-conditioned systems seen in truss modeling.

For weakly nonlinear problems where the global stiffness matrix does not drastically change throughout the simulation (e.g. the hole-in-plate in Section 4.5), we found that using the inverse of the *undeformed* stiffness matrix as a preconditioner for a nonlinear iterative solver — namely, the ngmres solver — proved to be an efficient solving technique⁶. This is achieved by computing the cholesky decomposition of the original undeformed stiffness matrix \mathbf{K}_0

³PETSc runtime options: "-pc_type redistribute -redistribute_pc_type cholesky -redistribute_pc_factor_mat_solver_type mumps"

⁴PETSc runtime options: "-pc_type redistribute -redistribute_pc_type fieldsplit -redistribute_pc_fieldsplit_factor_mat_solver_type mumps" -pc_fieldsplit_0_fields "indices_of_translational_dofs" -pc_fieldsplit_1_fields "indices_of_rotational_dofs"

⁵PETSc runtime options: "-pc_type gamg"

⁶PETSc runtime options: "-snes_type ngmres -npc_snes_type newtonls -npc_snes_lag_jacobian 100000 -snes_lag_jacobian_persists -npc_pc_type redistribute -npc_redistribute_pc_type cholesky -npc_redistribute_pc_factor_mat_solver_type mumps"

once, then using the modified residual $\tilde{\mathbf{F}} = \mathbf{K}_0^{-1}(\mathbf{F})$ as the input to the non-linear solver for the rest of the simulation.

For extremely nonlinear problems (e.g. the indentation of the 3D kagome lattice in Section 4.3), we found that Newton’s method paired a direct LU solver provided by MUMPS and the line search algorithm of Moré and Thunent [75] in the TAO package provided a very robust solution strategy where most other solvers seemed to fail to converge. This combination of direct solver and robust line search was able to deal with the buckling of beams, and provides the guarantee that the converged solution is a minimum, rather than an unstable stationary point, of the total potential energy.

6.1.6 Remeshing and Updating State

As mentioned in Section 3.5, we are able to refine our mesh if it undergoes severe deformation by marking elements for refinement, then placing new repUCs at the nearest valid lattice site to the midpoint of the longest edge of the element. However, we do not have the ability to manipulate the existing connectivity to incorporate the new repUC locations, so we delete all connectivity, remesh the entire geometry, rebuild the sampling elements, and then continue with the solve. Once the system is at equilibrium and no refinement has occurred, the state of the system is updated. In the case of dynamics, the previous displacements, velocities and accelerations are stored, and for plasticity, the new internal variables are stored. When simulating the brittle failure of beams, only a single beam is failed during this step, then the simulation re-enters the equilibrium-refinement loop (described in Algorithm 1). Once all of the internal variables are updated and the system is at equilibrium, the simulation progresses to the next load step.

6.2 Computational Costs

As mentioned in Section 3.4, the QC method reduced the computational cost of lattice simulations in two ways: it reduces 1) the number of dofs in the system, and 2) the cost of computing energetic quantities (energy, force, stiffness). Although related, both of these aspects reduce the computational cost (both memory and runtimes) in different ways.

6.2.1 Assembling the System

By choosing repUCs to parametrize the displacements of the entire lattice, the number of dofs in system is reduced by a factor of the repUC density⁷. The size of the global force vector is therefore reduced by a factor of repUC density. The overall dimensions of the global stiffness matrix are also reduced by a factor of repUC density, but the memory required by the sparse matrix depends on the number of *nonzeros* in the matrix, not the number of rows and column of the matrix. Depending on the summation rule being used, the number of nonzeros of the global stiffness matrix (related to the connectivity of the energy sampling elements defined in Section 6.1.2) can vary greatly.

For example, take the square truss lattice modeled with beam finite elements such that each node has 3 dofs. In the fully-resolved simulation, each unit cell on the interior of the domain is connected to exactly 4 other unit cells (up, down, left, and right). The maximum number of nonzero components in the rows corresponding to this unit cell is therefore 15 (the 3 components in the diagonal portion of the matrix plus the 4×3 components stemming from the truss elements connecting the unit cell to it's neighbors). However, if a nonlocal version of the QC method is used to model the square lattice, it is possible that a unit cell is directly coupled with more than just its 4 closest neighbors. If a sample unit cell location is chosen such that sample truss elements cross element boundaries, a single truss element could single handedly couple the unit cell to 4 other unit cells (e.g. the green sampling truss members in Fig. 6.2), resulting in more nonzeros per row than the fully-resolved simulation. This effect is even more pronounced in three dimensions and with higher-order interpolations. For the simulation of the tetrakaidecahedron lattice with distributed shear load in Section 5.3.4, the maximum number of nonzero entries in the stiffness matrix in a given row increased by a factor of 65 (from 15 to 975) when using the quadratic interpolation instead of full resolution.

Even though the number of nonzeros per row of the stiffness matrix can increase when the truss is coarse-grained, the increase is one that is relatively constant (i.e. it does not depend on *how coarse* the mesh is nor does it scale with the *number* of sampling unit cells). As a result, it is not obvious whether coarsening the truss by a small degree will increase or decrease the total number of nonzeros in the matrix. However, when a significantly low repUC density

⁷As a reminder, the repUC density is the ratio of the number of repUCs to total number of unit cells in the simulation: N_r/N_{UC} .

is used, the decrease in the number of rows of the outpaces the increase in the number of nonzeros per row, leading to an overall decrease in the number of nonzeros in the matrix.

On the other hand, the computational time it takes to assemble the quantities is directly tied to the efficiency of the summation rule. As seen in Algorithms 2 to 4, the assembly requires the sum over all of the sample unit cells (or equivalently, over the sample truss elements), and therefore, the cost of assembling these quantities is governed by the *sample unit cell density*⁸.

6.2.2 Solving the System

The reduction in computational cost of the QC method also can also greatly depend on the solver being that is used to minimize the energy.

For atomic lattice computations, it is common to use force-based solvers (e.g. steepest descent, nonlinear conjugate gradient, fast interial relaxation engine [23]). When using these types of solvers, the majority of the computation time is spent computing forces, and thus the overall run time of the simulation depends on the time it takes to assemble the global force vector and the number of iterations the solver needs to converge. As mentioned above, the time it takes to assemble of the global force vector scales linearly with the number of sampling unit cells (i.e. uses $\mathcal{O}(N_s)$ operations). The convergence of the solver on the other hand depends on the number of unknowns. In practice, iterative solvers can be used to achieve approximate solutions in roughly a constant number of iterations, with each iteration using $\mathcal{O}(N_s)$ operations to assemble the force vector and $\mathcal{O}(N_r)$ operations to update the solution vector. Usually, the time spent assembling the force vector dominates over the time spent updating the solution vector, and therefore the speed-up associated with force-based solvers depends highly on the sample unit cell density.

For truss lattice computations, unpreconditioned iterative methods like those commonly used in atomistics become unusable because the poorly conditioned nature of the problem results in terrible convergence properties. Instead, we use matrix-based solvers like such as Newton methods. When solving a problem with matrix-based solvers, the computational advantages of the QC method are realized in different ways.

⁸Similar to the repUC density, the sample unit cell density is the ratio of the number of sampling unit cells to total number of unit cells in the simulation: N_s/N_{UC} .

For instance, when using a Newton-Raphson iteration with a direct solver, the memory savings become more significant, especially with the global stiffness matrix. When the QC method is used with a sufficiently coarse mesh, the memory required to store the sparse global stiffness matrix is reduced. While the number of nonzeros per row can increase due to extra coupling of repUCs from sampling unit cells (but in general does not increase with the *number of sample unit cells*), the number of rows of the stiffness matrix (and therefore the total number of nonzeros) is governed by the number of repUCs. The number of repUCs therefore drives the memory required to store the global stiffness matrix, and more importantly, the memory required for the direct solver.

Since the time it takes to solve a linear system increases with an increase in the number of nonzeros in the matrix, the number of repUCs also drives the computation time of the linear solve. Unlike matrix-free methods, the direct linear solve usually dominates the simulation time since it typically takes longer to solve the system than it does to assemble it. This leads to the conclusion that the speed-up associated with the QC method when a Newton-Raphson solver is used will be more dependent on the repUC density. Of course, this depends on a reasonable summation rule where the number of sample unit cells is of the same order as the number of repUCs. If an exact summation rule is used in a very coarse simulation, then the time it takes to assemble the system could begin to dominate the computation time, and using a less expensive summation rule could significantly improve simulation times.

6.2.3 Conclusions

As discussed above, the computational savings associated with the QC method are complicated and depend heavily on the choice of interpolation and energy approximation rules, but also on the choice of solver being used. However, the following rule of thumb is useful in estimating the computational savings: If a solver is being used that relies heavily on force computations, the speed up will be closely related to the sample unit cell density, but if a solver that relies on linear solves is used, the speed up will be closely related to repUC density. Since most of the truss lattice problems investigated in this thesis are ill-conditioned, we relied heavily on matrix-based solvers, and thus the computational savings were closely related to the repUC density.

Even though the exact benefits of the QC method are nuanced, we give the

	Fully-Resolved	Coarse-Grained		
RepUC Density	1	$(\frac{1}{2})^3$	$(\frac{1}{3})^3$	$(\frac{1}{6})^3$
Assembly	8 sec	19 sec	7 sec	1.5 sec
Linear Solve	1282 sec	1931 sec	177 sec	8.5 sec
Total	1390 sec	1950 sec	184 sec	10 sec

Table 6.1: Time (in seconds) spent assembling and solving the lattice block shear simulations in Section 5.3.4 with quadratic interpolation. Simulations were performed on a machine with an Intel Xeon CPU E5-2650 processor using 12 cores with 1 MPI process per core.

time spent in different portions of the code for two of the example problems in this thesis in order to give the reader an idea of the computational speed-up in different scenarios. Namely, we look at

1. The block under distributed shear load in Section 5.3.4. As mentioned above, the linear system was solved by iteratively solving for the translational and rotational dofs separately using direct solvers (Cholesky decomposition) for each sub-solve. This is an example of a problem that relies heavily on direct linear solves, and less so on assembly of global force vectors and stiffness matrices.
2. The dynamic indentation of the triangle lattice with failure in Section 4.4. Each load step was solved using Newton’s method paired with an algebraic-multigrid-preconditioned gmres linear iterative solver and a line search. In contrast to the distributed shear simulation, this non-linear solving scheme relied heavily on force and stiffness computations. The algebraic multigrid preconditioner provided efficiently linear solves, and much of the time was spent performing line searches to determine the step size. The simulation also is a good example because it contains the additional cost of performing adaptive refinement.

The run times of the two simulations are in Table 6.1 and Table 6.2.

For the lattice block shearing simulations in Section 5.3.4, the run time of the simulation depends heavily on the repUC density. The lattice that is coarsened by a factor of 2 in each direction (leading to a repUC density of $1/2^3$) actually takes twice as long to assemble the global stiffness matrix, and roughly 1.5 times as long to solve the system. This is due to the fact that the number of

	Fully-Resolved	Coarse-Grained
Assembly	3296 sec	1312 sec
Linear Solve	1744 sec	292 sec
Total Solve	5040 sec	1604 sec
Remeshing	N/A	289 sec

Table 6.2: Time (in seconds) spent in different parts of the code for the dynamic indentation simulations in Section 4.4. "Total Solve" represents the total time in the nonlinear solver (i.e. assembling systems, solving linear systems, and performing line searches). Simulations were performed serially on a machine with an Intel Xeon CPU E5-2650 processor.

nonzeros per row significantly increases when coarsening the mesh, requiring a larger linear system to be assembled and solved. However, when the mesh is coarsened enough, the stiffness matrix becomes much smaller than the fully-resolved system and can be solved extremely quickly. For the coarsest mesh, the simulation took less than 1% of the time of the fully-resolved simulation.

However, for the dynamic indentation simulations, most of the time was spent assembling the system. Since the sampling unit cell density is always greater than the repUC density, the time spent in the linear solve portion of the simulation was reduced more than the time spent in assembly. Furthermore, we see that the cost associated with global remeshing and rebuilding all of the sampling unit cells was a small portion of the run time in the serial simulation.

DISCUSSION AND OUTLOOK

7.1 Accomplishments

We have extended the fully-nonlocal quasicontinuum method to describe multilattice truss topologies in 2D and 3D by introducing a new interpolation scheme by which the dofs of each Bravais lattice in the multilattice are retained at the macroscale and are interpolated in coarse-grained regions. We have also introduced geometrically nonlinear deformation into the truss QC methodology by using corotational beams for the description of individual truss members. A natural extension of the optimal summation rule of Amelang et al. [3] to multilattice systems was used to efficiently sample the energy of the coarse-grained lattices. We introduced the concept of stretch locking — a phenomenon analogous to volumetric locking in finite elements that causes the QC method to significantly overpredict the stiffness of bending-dominate multilattices — as well as proposed a solution to the problem by using quadratic interpolation. The method was numerically implemented in a code with adaptive mesh refinement to automatically improve the resolution of the QC discretization where necessary and to capture localized deformation.

We have tested this framework’s ability to correctly model the effective elastic properties of various 2D and 3D lattices by uniformly deforming coarse-grained truss lattice blocks and comparing the effective elastic constants to those predicted by homogenization theory. When using an exact summation rule (where every unit cell in the simulation is a sampling unit cell), we found that the interpolation was able to exactly match the homogenization theory for all coarsened meshes. When the interpolation was paired with the optimal summation rule on randomly coarsened samples, small, acceptable errors were present in the predicted behavior in moderately coarsened meshes due to energy sampling errors that arise in all known sampling rules for the QC method.

Furthermore, we tested the new truss QC theory and the adaptive refinement routine with more complex deformations by performing coarse-grained versions of Boundary Layer Method simulations, which were previously used to

predict the brittle fracture toughness of various lattice topologies. The QC method with affine shape functions performs excellently for the triangle and kagome lattices; the maximum stress near the notch tip and total energy of the system matches the fully-resolved simulations within a few percent, while using only a small fraction of the dofs. The hexagon and star-shaped lattices did not show such agreement with the fully-resolved simulations when affine interpolation was used, leading to the discovery and subsequent investigation of stretch locking. Once quadratic interpolation was determined to mitigate stretch locking, the coarse-grained Boundary Layer Method simulations of the hexagon and star-shaped lattices were repeated using quadratic interpolation, and showed excellent agreement with the fully-resolved simulations.

We tested the geometric nonlinear capabilities of the new methods with indentation simulations on 2D and 3D kagome multilattices with nonlinear corrotational beams to model individual truss members, and adaptive refinement to improve the mesh as the deformation becomes severe. We found that both the inclusion of geometric nonlinearity and adaptive mesh refinement was necessary to match the behavior seen in the fully-resolved nonlinear simulations. Most importantly, we found that the coarse-grained simulations matched the fully-resolved simulations well while only using a fraction of the dofs.

We tested the QC methodology with material nonlinearity by performing tensile simulations of an elasto-plastic octet truss plate with a cylindrical hole in the center. We found that when using the higher-order summation rule to better approximate the energy near the boundaries of the domain, the coarse-grained methods were able to approximate both the global force-displacement behavior as well as the local plasticity near the cylindrical hole.

We investigated the dynamic response of truss lattice by simulating the dynamic rupture of an elastic-brittle triangle truss lattice when impacted by a cylindrical indenter. The overall behavior of the coarse-grained simulation matched the fully-resolve simulation, including the supershear rupture speed of the propagating crack.

We used Cook’s membrane, a common volumetric locking benchmark test used in finite elements, to investigate the newly discovered stretch locking that takes place in bending-dominated multilattices. We found that stretch locking can be mitigated by using quadratic interpolation rather than linear simplex elements, and outlined a method to create higher-order meshes with

consistent interpolation along with a set of summation rules in 2D and 3D that can be used to approximate the energy.

7.2 Future Work

Although much has been accomplished in this thesis, there are still many areas of research that would be useful in maturing the QC method for truss lattices and increase its applicability. Some of these areas include:

- **Dynamics** As shown in Section 4.4, the coarse-graining techniques of the QC method produce spurious elastic wave reflections due to the non-uniformity of the mesh in dynamic problems. For methods where there is a distinction between fully-resolved regions and coarse-grained regions, it is possible to artificially dampen high spatial-frequency waves at the boundary of the regions to mitigate the elastic waves from reflecting back into the fully-resolved region. However, for the fully-nonlocal version of the QC method, where no such distinction is made, it is not clear how to solve this issue, and would be a good topic for future research. Of course, if the material that the truss is fabricated out of has high damping, then when damping is included in the model, the high-frequency waves would decay relatively fast, and the wave reflections would cause less of an issue.

Another limitation of the dynamics formulation presented in this thesis is related to the time scale needed to resolve high-frequency vibrations. For multiscale truss lattices, the period of oscillation of a single truss member can be many orders of magnitude smaller than the time scales related to external loading. While a suitable implicit dynamics scheme can be unconditionally stable even for large time steps, a small Δt is needed to model vibrations in the truss lattice. Kulkarni et al. [62] introduced a remedy to this issue for modeling atomic lattices (called *hotQC*), by introducing entropy and temperature variables at each atomic site to describe the high-frequency vibrations of each atom. Similar techniques could be applied to truss lattices to model e.g. the mechanical response of a truss lattice in a sandwich core on a launch vehicle or airplane, where significant vibrations could effect the behavior of the lattice.

- **Summation Rules** Unlike the finite element method which uses Gaussian quadrature with its guarantees for integrating the energy *exactly*

for polynomials up to a certain degree, the QC method has no equivalent summation rule. This has resulted in many ad-hoc rules (e.g. [3, 17, 37, 49, 58]) that all have significant flaws. As discussed in Section 3.4.2, almost all of these rules are not 0th-order accurate, and introduce spurious force artifacts even in simple configurations (undeformed configuration for atomic lattices, and affinely deformed configurations for truss lattices). Developing summation rules with similar guarantees to Gaussian quadrature could be impractical for atomic simulations. Both the nonlinearity and the large number of interactions stemming from the long-range nature of inter-atomic potentials intuitively would seem to require a large number of sampling interactions to ensure such a guarantee. Since force-based solvers are typically used to solve for equilibrium of atomic ensembles, the large number of sample interactions would most likely be prohibitively expensive. However, for truss lattice with their short-range interactions and relatively few neighbors, it may be possible to develop summation rules that offer similar guarantees to Gaussian quadrature without being prohibitively expensive. Furthermore, if the system is solved with matrix-based solvers, the increase in computational cost stemming from a more accurate summation rule may not hinder the performance of the method by a significant amount. Beex et al. [17] outlined a summation rule that exactly reproduced the energy of a lattice, but it was still expensive (all atoms within a certain distance of the element boundaries were taken as sample atoms), and only works for affine shape functions. Further work on summation rules can be used to reduce the error associated with energy approximation, especially in regions near fully-resolved sections.

- **Goal-oriented Refinement** The adaptive refinement techniques used in this thesis are ad-hoc. The second-invariant refinement metric was used because it was expected that large shear deformations would occur in the regions of interest (e.g. localized shear near the notch tip in Section 4.2 and buckling near the indenter in Section 4.3), and the maximum elongation criterion was used in Section 4.4 such that the QC mesh would be fully-resolved to capture individual elements breaking. Goal-oriented mesh adaptation stemming from a posteriori error approximation (e.g. [6, 68]) could be used to provide a more robust refinement technique.

- **Local Refinement** Due to Amdahl’s law¹, the computational scalability of the fully-nonlocal adaptive QC method depends on the ability to locally refine the mesh without having to regenerate the global mesh (and subsequently, all of the sampling unit cells). Tembhekar et al. [108] introduced a local refinement algorithm based on longest-edge bisection, but was only implemented in two dimensions. As mentioned in Tembhekar et al. [108], the extension to three dimensions requires a constrained 3D mesh generator that is not currently available.
- **Mesh Coarsening** From the perspective of computational efficiency, the ability to coarsen the mesh in uninteresting regions is just as important as refining the mesh in interesting regions. In particular, for dynamic simulations, it may be possible to refine the mesh in front of an elastic wave and coarsen the mesh behind it to retain accuracy of the simulation without having to fully-resolve the entire simulation.
- **Sophisticated Truss Models** These simulations in this thesis were the first QC simulations to utilize geometrically nonlinear beams to model buckling of truss members. However, the structural finite element models used here have their limitations. For example, while the corotational framework defined in Section 3.2.2 can be paired with any underlying beam element that depends on the axial strain and local rotations of the nodes, we have restricted ourselves to model either linear elastic or brittle Euler-Bernoulli beams. In order to model more complex material behavior (e.g. nonlinear elastic polymers, metal plasticity, etc.), more sophisticated underlying beam elements are needed (e.g. [63]).

Additionally, as shown in Meza et al. [71], Portela et al. [81], many of the hollow or non-slender truss lattices that are currently being manufactured cannot be accurately approximated with structural finite elements. Higher-order effects coming from finite-sized nodes begin to appear in truss lattices with higher relative density. It is possible that reduced order node models like those developed in Portela et al. [81] could be

¹Amdahl’s law is a formula that can be used to predict the theoretical speed-up of parallel program, but it is commonly used to show that the run time of a partially-parallel code is bounded by above by the run time of the serial portion of the code, regardless of the speed-up in the parallel section. In this context, it refers to the fact that the need to perform global mesh refinement serially hinders the speed-up of the otherwise parallel code.

used in conjunction with structural finite elements in the QC method to provide a more accurate description of the underlying truss lattice.

- **Contact** As outlined in Gibson and Ashby [48], there are three distinct regimes in the stress-strain response of cellular solids undergoing compression: 1) initial *linear elastic* response, 2) a long *collapse plateau* where the stress remains relatively constant due to buckling or yielding of the microstructure, and 3) *densification* where the solid dramatically stiffens due to contact of the microstructure. The QC formulation in this thesis is able to model the first two regimes by including material and geometric nonlinearity, but it is unable to capture densification of the truss lattice due to inter-element contact. Since periodic truss lattices are ideal candidates for absorbing large amount of energy in case of impact, contact between truss members must be taken into account to accurately model the large compressive strains and densification that would be seen in impact scenarios.

Overall, the research presented in this thesis extends the quasicontinuum method in various directions. The fully-nonlocal multilattice formulation establishes a novel method for modeling periodic truss multilattices using a wide variety of models for truss members. The framework developed here can simulate the nonlinear mechanical response of large periodic truss lattices in an efficient manner, and can be used to aid in the design of new metamaterials.

BIBLIOGRAPHY

- [1] I. Quintana Alonso and N.A. Fleck. Compressive response of a sandwich plate containing a cracked diamond-celled lattice. *Journal of the Mechanics and Physics of Solids*, 57(9):1545 – 1567, 2009. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2009.05.008>. URL <http://www.sciencedirect.com/science/article/pii/S0022509609000763>.
- [2] J. Amelang. *A Fully-Nonlocal Energy-based Formulation and High-performance Realization of the Quascontinuum Method*. PhD thesis, California Institute of Technology, 9 2016. URL <http://resolver.caltech.edu/CaltechTHESIS:09152015-212147583>.
- [3] J.S. Amelang, G.N. Venturini, and D.M. Kochmann. Summation rules for a fully nonlocal energy-based quascontinuum method. *Journal of the Mechanics and Physics of Solids*, 82:378 – 413, 2015. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2015.03.007>. URL <http://www.sciencedirect.com/science/article/pii/S0022509615000630>.
- [4] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L’Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.
- [5] P. R. Amestoy, A. Guermouche, J.-Y. L’Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.
- [6] Marcel Arndt and Mitchell Luskin. Goal-oriented atomistic-continuum adaptivity for the quascontinuum approximation. *International Journal for Multiscale Computational Engineering*, 5(5):407–415, 2007. ISSN 1543-1649.
- [7] Takashi Asada, Yuji Tanaka, and Nobutada Ohno. Two-scale and full-scale analyses of elastoplastic honeycomb blocks subjected to flat-punch indentation. *International Journal of Solids and Structures*, 46(7):1755 – 1763, 2009. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijssolstr.2008.12.024>. URL <http://www.sciencedirect.com/science/article/pii/S0020768309000122>.
- [8] S. Balay, W. D. Gropp, L. Curfman McInnes, and B. E. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

- [9] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. Curfman McInnes, K. Rupp, B. F. Smith, S. Zampini, Zhang H., and H. Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2017. URL <http://www.mcs.anl.gov/petsc>.
- [10] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. Curfman McInnes, K. Rupp, B. F. Smith, S. Zampini, Zhang H., and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017. URL <http://www.mcs.anl.gov/petsc>.
- [11] Jean-Marc Battini and Costin Pacoste. On the choice of the linear element for corotational triangular shells. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6362 – 6377, 2006. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2006.01.007>. URL <http://www.sciencedirect.com/science/article/pii/S0045782506000508>.
- [12] J. Bauer, A. Schroer, R. Schwaiger, and O. Kraft. Approaching theoretical strength in glassy carbon nanolattices. *Nature Materials*, 15:438 EP –, 02 2016. URL <https://doi.org/10.1038/nmat4561>.
- [13] Z. P. Bažant. Spurious reflection of elastic waves in nonuniform finite element grids. *Computer Methods in Applied Mechanics and Engineering*, 16(1):91 – 100, 1978. ISSN 0045-7825. doi: [https://doi.org/10.1016/0045-7825\(78\)90035-X](https://doi.org/10.1016/0045-7825(78)90035-X). URL <http://www.sciencedirect.com/science/article/pii/004578257890035X>.
- [14] Z. P. Bažant and Z. Celep. Spurious reflection of elastic waves in nonuniform meshes of constant and linear strain unite elements. *Computers Structures*, 15(4):451 – 459, 1982. ISSN 0045-7949. doi: [https://doi.org/10.1016/0045-7949\(82\)90080-3](https://doi.org/10.1016/0045-7949(82)90080-3). URL <http://www.sciencedirect.com/science/article/pii/0045794982900803>.
- [15] L. A. A. Beex, R. H. J. Peerlings, and M. G. D. Geers. A quasicontinuum methodology for multiscale analyses of discrete microstructural models. *International Journal for Numerical Methods in Engineering*, 87(7):701–718, 2011. ISSN 1097-0207. doi: 10.1002/nme.3134. URL <http://dx.doi.org/10.1002/nme.3134>.
- [16] L.A.A. Beex, P. Kerfriden, T. Rabczuk, and S.P.A. Bordas. Quasicontinuum-based multiscale approaches for plate-like beam lattices experiencing in-plane and out-of-plane deformation. *Computer Methods in Applied Mechanics and Engineering*, 279:348 – 378, 2014. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2014.06.018>. URL <http://www.sciencedirect.com/science/article/pii/S0045782514002047>.

- [17] L.A.A. Beex, R.H.J. Peerlings, and M.G.D. Geers. Central summation in the quasicontinuum method. *Journal of the Mechanics and Physics of Solids*, 70:242 – 261, 2014. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2014.05.019>. URL <http://www.sciencedirect.com/science/article/pii/S0022509614001100>.
- [18] L.A.A. Beex, R.H.J. Peerlings, and M.G.D. Geers. A multiscale quasicontinuum method for lattice models with bond failure and fiber sliding. *Computer Methods in Applied Mechanics and Engineering*, 269:108 – 122, 2014. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2013.10.027>. URL <http://www.sciencedirect.com/science/article/pii/S004578251300279X>.
- [19] L.A.A. Beex, R.H.J. Peerlings, and M.G.D. Geers. A multiscale quasicontinuum method for dissipative lattice models and discrete networks. *Journal of the Mechanics and Physics of Solids*, 64:154 – 169, 2014. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2013.11.010>. URL <http://www.sciencedirect.com/science/article/pii/S0022509613002445>.
- [20] L.A.A. Beex, O. Rokoš, J. Zeman, and S.P.A. Bordas. Higher-order quasicontinuum methods for elastic and dissipative lattice models: uniaxial deformation and pure bending. *GAMM-Mitteilungen*, 38(2):344–368, 2015. doi: 10.1002/gamm.201510018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201510018>.
- [21] A. Bensoussan, J.L. Lions, and G. Papanicolaou. *Asymptotic Analysis for Periodic Structures*. Studies in Mathematics and Its Applications. North-Holland Publishing Company, 1978. ISBN 9780444851727. URL <http://books.google.com/books?id=q-Gvuk7fXWcC>.
- [22] J B Berger, H N G Wadley, and R M McMeeking. Mechanical metamaterials at the theoretical limit of isotropic elastic stiffness. *Nature*, 543: 533, feb 2017. URL <http://dx.doi.org/10.1038/nature21075><http://10.0.4.14/nature21075><https://www.nature.com/articles/nature21075#supplementary-information>.
- [23] Erik Bitzek, Pekka Koskinen, Franz Gähler, Michael Moseler, and Peter Gumbsch. Structural relaxation made simple. *Phys. Rev. Lett.*, 97: 170201, Oct 2006. doi: 10.1103/PhysRevLett.97.170201. URL <https://link.aps.org/doi/10.1103/PhysRevLett.97.170201>.
- [24] Michel Bouchon, Marie-Paule Bouin, Hayrullah Karabulut, M. Nafi Toksös, Michel Dietrich, and Ares J. Rosakis. How fast is rupture during an earthquake? new insights from the 1999 turkey earthquakes. *Geophysical Research Letters*, 28(14):2723–2726, 2001. doi: 10.1029/2001GL013112. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001GL013112>.

- [25] Dávid Bozsaky. The historical development of thermal insulation materials. *Periodica Polytechnica Architecture*, 41(2):49–56. doi: <https://doi.org/10.3311/pp.ar.2010-2.02>. URL <https://pp.bme.hu/ar/article/view/12>.
- [26] P. Brune, M. Knepley, B. Smith, and X. Tu. Composing scalable nonlinear algebraic solvers. *SIAM Review*, 57(4):535–565, 2015. doi: [10.1137/130936725](https://doi.org/10.1137/130936725). URL <https://doi.org/10.1137/130936725>.
- [27] William Bryan. Memory foam, Feb 2016. URL <https://www.nasa.gov/offices/oct/40-years-of-nasa-spinoff/memory-foam>.
- [28] J.Y. Chen, Y. Huang, and M. Ortiz. Fracture analysis of cellular materials: A strain gradient model. *Journal of the Mechanics and Physics of Solids*, 46(5):789 – 828, 1998. ISSN 0022-5096. doi: [http://dx.doi.org/10.1016/S0022-5096\(98\)00006-4](http://dx.doi.org/10.1016/S0022-5096(98)00006-4). URL <http://www.sciencedirect.com/science/article/pii/S0022509698000064>.
- [29] Sukjoo Choi and Bhavani V. Sankar. A micromechanical method to predict the fracture toughness of cellular materials. *International Journal of Solids and Structures*, 42(5):1797 – 1817, 2005. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2004.08.021>. URL <http://www.sciencedirect.com/science/article/pii/S0020768304004676>.
- [30] D. Cioranescu and P. Donato. *An Introduction to Homogenization*. Oxford lecture series in mathematics and its applications. Oxford University Press, 1999. ISBN 9780198565536. URL <https://books.google.com/books?id=bggzAAAAAAAJ>.
- [31] M.A. Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Computer Methods in Applied Mechanics and Engineering*, 81(2):131 – 150, 1990. ISSN 0045-7825. doi: [https://doi.org/10.1016/0045-7825\(90\)90106-V](https://doi.org/10.1016/0045-7825(90)90106-V). URL <http://www.sciencedirect.com/science/article/pii/004578259090106V>.
- [32] M.A. Crisfield. *Non-linear finite element analysis of solids and structures*. Number v. 2 in Non-linear Finite Element Analysis of Solids and Structures. Wiley, 1997. ISBN 9780471956495. URL <https://books.google.ch/books?id=INtRAAAAMAAJ>.
- [33] Xiaodong Cui, Zhenyu Xue, Yongmao Pei, and Daining Fang. Preliminary study on ductile fracture of imperfect lattice materials. *International Journal of Solids and Structures*, 48(25):3453 – 3461, 2011. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2011.08.013>. URL <http://www.sciencedirect.com/science/article/pii/S0020768311003003>.

- [34] V.S. Deshpande, M.F. Ashby, and N.A. Fleck. Foam topology: bending versus stretching dominated architectures. *Acta Materialia*, 49(6):1035 – 1040, 2001. ISSN 1359-6454. doi: [https://doi.org/10.1016/S1359-6454\(00\)00379-7](https://doi.org/10.1016/S1359-6454(00)00379-7). URL <http://www.sciencedirect.com/science/article/pii/S1359645400003797>.
- [35] A. Desmoulins and D.M. Kochmann. Local and nonlocal continuum modeling of inelastic periodic networks applied to stretching-dominated trusses. *Computer Methods in Applied Mechanics and Engineering*, 313:85 – 105, 2017. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2016.09.027>. URL <http://www.sciencedirect.com/science/article/pii/S0045782516301566>.
- [36] M. Dobson, R. S. Elliott, M. Luskin, and E. B. Tadmor. A multilattice quasicontinuum for phase transforming materials: Cascading cauchy born kinematics. *Journal of Computer-Aided Materials Design*, 14(1): 219–237, Dec 2007. ISSN 1573-4900. doi: 10.1007/s10820-007-9084-7. URL <https://doi.org/10.1007/s10820-007-9084-7>.
- [37] B. Eidel and A. Stukowski. A variational formulation of the quasicontinuum method based on energy sampling in clusters. *J. Mech. Phys. Solids*, 57:87–108, 2009.
- [38] F. Feyel. A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in Applied Mechanics and Engineering*, 192(28):3233 – 3244, 2003. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(03\)00348-7](https://doi.org/10.1016/S0045-7825(03)00348-7). URL <http://www.sciencedirect.com/science/article/pii/S0045782503003487>. Multiscale Computational Mechanics for Materials and Structures.
- [39] F. Feyel and J.-L. Chaboche. Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3):309 – 330, 2000. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(99\)00224-8](https://doi.org/10.1016/S0045-7825(99)00224-8). URL <http://www.sciencedirect.com/science/article/pii/S0045782599002248>.
- [40] N. A. Fleck, V. S. Deshpande, and M. F. Ashby. Micro-architected materials: past, present and future. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2121): 2495–2516, 2010. doi: 10.1098/rspa.2010.0215. URL <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2010.0215>.
- [41] Norman A. Fleck and XinMing Qiu. The damage tolerance of elastic–brittle, two-dimensional isotropic lattices. *Journal of the Mechanics and Physics of Solids*, 55(3):562 – 588, 2007. ISSN 0022-5096.

- doi: <https://doi.org/10.1016/j.jmps.2006.08.004>. URL <http://www.sciencedirect.com/science/article/pii/S0022509606001359>.
- [42] L. B. Freund. The mechanics of dynamic shear crack propagation. *Journal of Geophysical Research: Solid Earth*, 84(B5):2199–2209, 1979. doi: 10.1029/JB084iB05p02199. URL <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB084iB05p02199>.
 - [43] L. B. Freund and R. J. Clifton. On the uniqueness of plane elastodynamic solutions for running cracks. *Journal of Elasticity*, 4(4):293–299, Dec 1974. ISSN 1573-2681. doi: 10.1007/BF00048612. URL <https://doi.org/10.1007/BF00048612>.
 - [44] J. Clerk Maxwell F.R.S. L. on the calculation of the equilibrium and stiffness of frames. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 27(182):294–299, 1864. doi: 10.1080/14786446408643668. URL <https://doi.org/10.1080/14786446408643668>.
 - [45] M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. Gradient-enhanced computational homogenization for the micro-macro scale transition. *Journal de Physique IV, Colloque*, 11(5):145–152, 2001. ISSN 1155-4339. doi: 10.1051/jp4:2001518.
 - [46] G. Geymonat, S. Müller, and N. Triantafyllidis. Homogenization of nonlinearly elastic materials, microscopic bifurcation and macroscopic loss of rank-one convexity. *Archive for Rational Mechanics and Analysis*, 122:231–290, 1993. doi: 10.1007/BF00380256. URL <http://dx.doi.org/10.1007/BF00380256>.
 - [47] Somnath Ghosh, Kyunghoon Lee, and Prasanna Raghavan. A multi-level computational model for multi-scale damage analysis in composite and porous materials. *International Journal of Solids and Structures*, 38(14):2335 – 2385, 2001. ISSN 0020-7683. doi: [https://doi.org/10.1016/S0020-7683\(00\)00167-0](https://doi.org/10.1016/S0020-7683(00)00167-0). URL <http://www.sciencedirect.com/science/article/pii/S0020768300001670>.
 - [48] L.J. Gibson and M.F. Ashby. *Cellular Solids: Structure and Properties*. Cambridge Solid State Science Series. Cambridge University Press, 1997. ISBN 9780521495608. URL <https://books.google.com/books?id=3wW8bwAACAAJ>.
 - [49] M. Gunzburger and Y. Zhang. A quadrature-rule type approximation to the quasi-continuum method. *Multiscale Model. Simul.*, 8(2):571–590, 2010.
 - [50] Axel S. Herrmann, Pierre C. Zahlen, and Ichwan Zuardy. Sandwich structures technology in commercial aviation. In O.T. Thomsen,

- E. Bozhevolnaya, and A. Lyckegaard, editors, *Sandwich Structures 7: Advancing with Sandwich Structures and Materials*, pages 13–26, Dordrecht, 2005. Springer Netherlands. ISBN 978-1-4020-3848-8.
- [51] Susan Hert and Michael Seel. dD convex hulls and delaunay triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.14 edition, 2019. URL <https://doc.cgal.org/4.14/Manual/packages.html#PkgConvexHullD>.
 - [52] R. Hill. Elastic properties of reinforced solids: Some theoretical principles. *Journal of the Mechanics and Physics of Solids*, 11(5):357–372, 1963. doi: 10.1016/0022-5096(63)90036-X. URL <http://www.sciencedirect.com/science/article/pii/002250966390036X>.
 - [53] J.S. Huang and L.J. Gibson. Fracture toughness of brittle honeycombs. *Acta Metallurgica et Materialia*, 39(7):1617 – 1626, 1991. ISSN 0956-7151. doi: [https://doi.org/10.1016/0956-7151\(91\)90249-Z](https://doi.org/10.1016/0956-7151(91)90249-Z). URL <http://www.sciencedirect.com/science/article/pii/095671519190249Z>.
 - [54] R.G. Hutchinson and N.A. Fleck. The structural performance of the periodic truss. *Journal of the Mechanics and Physics of Solids*, 54(4):756 – 782, 2006. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2005.10.008>. URL <http://www.sciencedirect.com/science/article/pii/S002250960500205X>.
 - [55] Alan J. Jacobsen, Sky Mahoney, William B. Carter, and Steven Nutt. Vitreous carbon micro-lattice structures. *Carbon*, 49(3):1025 – 1032, 2011. ISSN 0008-6223. doi: <https://doi.org/10.1016/j.carbon.2010.10.059>. URL <http://www.sciencedirect.com/science/article/pii/S0008622310008195>.
 - [56] F. López Jiménez and N. Triantafyllidis. Buckling of rectangular and hexagonal honeycomb under combined axial compression and transverse shear. *International Journal of Solids and Structures*, 50(24):3934 – 3946, 2013. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2013.08.001>. URL <http://www.sciencedirect.com/science/article/pii/S0020768313003089>.
 - [57] C. Kane, J. E. Marsden, M. Ortiz, and M. West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering*, 49(10):1295–1325, 2000. doi: 10.1002/1097-0207(20001210)49:10<1295::AID-NME993>3.0.CO;2-W. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0207%2820001210%2949%3A10%3C1295%3A%3AAID-NME993%3E3.0.CO%3B2-W>.

- [58] J. Knap and M. Ortiz. An analysis of the quasicontinuum method. *J. Mech. Phys. Solids*, 49(9):1899–1923, September 2001.
- [59] D. M. Kochmann and J. A. Amelang. The quasicontinuum method: theory and applications. In C. R. Weinberger and G. J. Tucker, editors, *Multiscale Materials Modeling for Nanomechanics*, pages 159–195. Springer, New York, 2016.
- [60] Dennis M Kochmann and Gabriela N Venturini. A meshless quasicontinuum method based on local maximum-entropy interpolation. *Modelling and Simulation in Materials Science and Engineering*, 22(3):034007, apr 2014. doi: 10.1088/0965-0393/22/3/034007. URL <https://doi.org/10.1088/0965-0393/22/3/034007>.
- [61] V.G. Kouznetsova, M.G.D. Geers, and W.A.M. Brekelmans. Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy. *Computer Methods in Applied Mechanics and Engineering*, 193(48):5525 – 5550, 2004. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2003.12.073>. URL <http://www.sciencedirect.com/science/article/pii/S0045782504002853>. Advances in Computational Plasticity.
- [62] Yashashree Kulkarni, Jaroslaw Knap, and Michael Ortiz. A variational approach to coarse graining of equilibrium and non-equilibrium atomistic description at finite temperature. *Journal of the Mechanics and Physics of Solids*, 56(4):1417 – 1449, 2008. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2007.09.005>. URL <http://www.sciencedirect.com/science/article/pii/S0022509607001834>.
- [63] Veronique Le Corvec. *Nonlinear 3d frame element with multi-axial coupling under consideration of local effects*. PhD thesis, University of California, Berkeley, 2012. URL <https://escholarship.org/uc/item/0sw164c3>.
- [64] Fabian Lipperman, Michael Ryvkin, and Moshe B. Fuchs. Fracture toughness of two-dimensional cellular material with periodic microstructure. *International Journal of Fracture*, 146(4):279–290, Aug 2007. ISSN 1573-2673. doi: 10.1007/s10704-007-9171-5. URL <https://doi.org/10.1007/s10704-007-9171-5>.
- [65] A.E.H. Love. *Mathematical Theory of Elasticity*. Cambridge Press, MA, 1893.
- [66] Shoji Maruo, Osamu Nakamura, and Satoshi Kawata. Three-dimensional microfabrication with two-photon-absorbed photopolymerization. *Opt. Lett.*, 22(2):132–134, Jan 1997. doi: 10.1364/OL.22.000132. URL <http://ol.osa.org/abstract.cfm?URI=ol-22-2-132>.

- [67] Karel Matouš, Marc G.D. Geers, Varvara G. Kouznetsova, and Andrew Gillman. A review of predictive nonlinear theories for multi-scale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192 – 220, 2017. ISSN 0021-9991. doi: <https://doi.org/10.1016/j.jcp.2016.10.070>. URL <http://www.sciencedirect.com/science/article/pii/S0021999116305782>.
- [68] Arash Memarnahavandi, Fredrik Larsson, and Kenneth Runesson. A goal-oriented adaptive procedure for the quasi-continuum method with cluster approximation. *Computational Mechanics*, 55(4):617–642, Apr 2015. ISSN 1432-0924. doi: 10.1007/s00466-015-1127-4. URL <https://doi.org/10.1007/s00466-015-1127-4>.
- [69] Lucas R. Meza, Satyajit Das, and Julia R. Greer. Strong, lightweight, and recoverable three-dimensional ceramic nanolattices. *Science*, 345(6202):1322–1326, 2014. ISSN 0036-8075. doi: 10.1126/science.1255908. URL <http://science.sciencemag.org/content/345/6202/1322>.
- [70] Lucas R. Meza, Alex J. Zelhofer, Nigel Clarke, Arturo J. Mateos, Dennis M. Kochmann, and Julia R. Greer. Resilient 3d hierarchical architected metamaterials. *Proceedings of the National Academy of Sciences*, 112(37):11502–11507, 2015. ISSN 0027-8424. doi: 10.1073/pnas.1509120112. URL <https://www.pnas.org/content/112/37/11502>.
- [71] Lucas R. Meza, Gregory P. Phlipot, Carlos M. Portela, Alessandro Maggi, Lauren C. Montemayor, Andre Comella, Dennis M. Kochmann, and Julia R. Greer. Reexamining the mechanical property space of three-dimensional lattice architectures. *Acta Materialia*, 140:424 – 432, 2017. ISSN 1359-6454. doi: <https://doi.org/10.1016/j.actamat.2017.08.052>. URL <http://www.sciencedirect.com/science/article/pii/S1359645417307073>.
- [72] Alexander Mielke and Tomas Roubicek. *Rate-Independent Systems - Theory and Application*, volume 193. 01 2015. ISBN 978-1-4939-2705-0. doi: 10.1007/978-1-4939-2706-7.
- [73] L C Montemayor, W H Wong, Y.-W. Zhang, and J R Greer. Insensitivity to Flaws Leads to Damage Tolerance in Brittle Architected Meta-Materials. *Scientific Reports*, 6:20570, feb 2016. URL <http://dx.doi.org/10.1038/srep20570><https://www.nature.com/articles/srep20570#supplementary-information>.
- [74] L.J. Mordell. *Diophantine Equations*. Monographs and textbooks in pure and applied mathematics. Academic Press, 1969. URL https://books.google.com/books?id=_5l6MwEACAAJ.

- [75] Jorge J. Moré and David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.*, 20:286–307, 1994.
- [76] D. Okumura, N. Ohno, and H. Noguchi. Elastoplastic microscopic bifurcation and post-bifurcation behavior of periodic cellular solids. *Journal of the Mechanics and Physics of Solids*, 52(3):641 – 666, 2004. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2003.07.002>. URL <http://www.sciencedirect.com/science/article/pii/S0022509603001248>.
- [77] M. Ortiz and L. Stainier. The variational formulation of viscoplastic constitutive updates. *Computer Methods in Applied Mechanics and Engineering*, 171(3):419 – 444, 1999. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(98\)00219-9](https://doi.org/10.1016/S0045-7825(98)00219-9). URL <http://www.sciencedirect.com/science/article/pii/S0045782598002199>.
- [78] M.R. O’Masta, L. Dong, L. St-Pierre, H.N.G. Wadley, and V.S. Deshpande. The fracture toughness of octet-truss lattices. *Journal of the Mechanics and Physics of Solids*, 98:271 – 289, 2017. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2016.09.009>. URL <http://www.sciencedirect.com/science/article/pii/S0022509616302952>.
- [79] Raj Kumar Pal, Massimo Ruzzene, and Julian J. Rimoli. A continuum model for nonlinear lattices under large deformations. *International Journal of Solids and Structures*, 96:300 – 319, 2016. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2016.05.020>. URL <http://www.sciencedirect.com/science/article/pii/S002076831630107X>.
- [80] S. Pellegrino and C.R. Calladine. Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures*, 22(4):409 – 428, 1986. ISSN 0020-7683. doi: [https://doi.org/10.1016/0020-7683\(86\)90014-4](https://doi.org/10.1016/0020-7683(86)90014-4). URL <http://www.sciencedirect.com/science/article/pii/0020768386900144>.
- [81] Carlos M. Portela, Julia R. Greer, and Dennis M. Kochmann. Impact of node geometry on the effective stiffness of non-slender three-dimensional truss lattice architectures. *Extreme Mechanics Letters*, 22:138 – 148, 2018. ISSN 2352-4316. doi: <https://doi.org/10.1016/j.eml.2018.06.004>. URL <http://www.sciencedirect.com/science/article/pii/S2352431618300725>.
- [82] I. Quintana Alonso and N. A. Fleck. Damage tolerance of an elastic-brittle diamond-celled honeycomb. *Scripta Materialia*, 56(8):693 – 696, 2007. ISSN 1359-6462. doi: <https://doi.org/10.1016/j.scriptamat.2006.12.027>. URL <http://www.sciencedirect.com/science/article/pii/S1359646207000152>.

- [83] I. Quintana-Alonso, S.P. Mai, N.A. Fleck, D.C.H. Oakes, and M.V. Twigg. The fracture toughness of a cordierite square lattice. *Acta Materialia*, 58(1):201 – 207, 2010. ISSN 1359-6454. doi: <https://doi.org/10.1016/j.actamat.2009.08.069>. URL <http://www.sciencedirect.com/science/article/pii/S135964540900593X>.
- [84] J. E. Reeve. On the volume of lattice polyhedra. *Proceedings of the London Mathematical Society*, s3-7(1):378–395, 1957. doi: 10.1112/plms/s3-7.1.378. URL <https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s3-7.1.378>.
- [85] B. Telgen R.N. Glaesener, C. Lestringant and D.M. Kochmann. Continuum models for stretching- and bending-dominated periodic trusses undergoing finite deformations. *International Journal of Solids and Structures*, 2019. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2019.04.022>. URL <http://www.sciencedirect.com/science/article/pii/S002076831930191X>.
- [86] O. Rokoš, L.A.A. Beex, J. Zeman, and R.H.J. Peerlings. A variational formulation of dissipative quasicontinuum methods. *International Journal of Solids and Structures*, 102-103:214 – 229, 2016. ISSN 0020-7683. doi: <https://doi.org/10.1016/j.ijsolstr.2016.10.003>. URL <http://www.sciencedirect.com/science/article/pii/S0020768316302943>.
- [87] O. Rokoš, R. H. J. Peerlings, J. Zeman, and L. A. A. Beex. An adaptive variational quasicontinuum methodology for lattice networks with localized damage. *International Journal for Numerical Methods in Engineering*, 112(2):174–200, 2017. doi: 10.1002/nme.5518. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5518>.
- [88] O. Rokoš, R.H.J. Peerlings, and J. Zeman. extended variational quasicontinuum methodology for lattice networks with damage and crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 320:769 – 792, 2017. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2017.03.042>. URL <http://www.sciencedirect.com/science/article/pii/S0045782516315997>.
- [89] Naomi E.R. Romijn and Norman A. Fleck. The fracture toughness of planar lattices: Imperfection sensitivity. *Journal of the Mechanics and Physics of Solids*, 55(12):2538 – 2564, 2007. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2007.04.010>. URL <http://www.sciencedirect.com/science/article/pii/S002250960700097X>.
- [90] A. J. Rosakis, O. Samudrala, and D. Coker. Cracks faster than the shear wave speed. *Science*, 284(5418):1337–1340, 1999. ISSN 0036-8075. doi: 10.1126/science.284.5418.1337. URL <https://science.sciencemag.org/content/284/5418/1337>.

- [91] M. Ryvkin and B. Nuller. Solution of quasi-periodic fracture problems by the representative cell method. *Computational Mechanics*, 20(1): 145–149, Jul 1997. ISSN 1432-0924. doi: 10.1007/s004660050231. URL <https://doi.org/10.1007/s004660050231>.
- [92] Ladan Salari-Sharif and L Valdevit. Accurate stiffness measurement of ultralight hollow metallic microlattices by laser vibrometry. *Experimental Mechanics*, 54:1491–1495, 10 2014. doi: 10.1007/s11340-014-9917-8.
- [93] T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, and W. B. Carter. Ultralight metallic microlattices. *Science*, 334(6058):962–965, 2011. doi: 10.1126/science.1211649. URL <http://www.sciencemag.org/content/334/6058/962.abstract>.
- [94] I. Schmidt and N.A. Fleck. Ductile fracture of two-dimensional cellular structures – dedicated to prof. dr.-ing. d. gross on the occasion of his 60th birthday. *International Journal of Fracture*, 111(4):327–342, Oct 2001. ISSN 1573-2673. doi: 10.1023/A:1012248030212. URL <https://doi.org/10.1023/A:1012248030212>.
- [95] Peter J. Schuster. Current trends in bumper design for pedestrian impact. 2006.
- [96] G. C. Sih, P. C. Paris, and G. R. Irwin. On cracks in rectilinearly anisotropic bodies. *International Journal of Fracture Mechanics*, 1(3): 189–203, Sep 1965. ISSN 1573-2673. doi: 10.1007/BF00186854. URL <https://doi.org/10.1007/BF00186854>.
- [97] J.C. Simo and T.J.R. Hughes. *Computational Inelasticity*. Interdisciplinary Applied Mathematics. Springer New York, 2000. ISBN 9780387975207. URL <https://books.google.com/books?id=ftL2AJL80PYC>.
- [98] L.I. Slepyan and M.V. Ayzenberg-Stepanenko. Some surprising phenomena in weak-bond fracture of a triangular lattice. *Journal of the Mechanics and Physics of Solids*, 50(8):1591 – 1625, 2002. ISSN 0022-5096. doi: [https://doi.org/10.1016/S0022-5096\(01\)00141-7](https://doi.org/10.1016/S0022-5096(01)00141-7). URL <http://www.sciencedirect.com/science/article/pii/S0022509601001417>.
- [99] V Sorkin, R S Elliott, and E B Tadmor. 22(5):055001, may 2014. doi: 10.1088/0965-0393/22/5/055001. URL <https://doi.org/10.1088%2F0965-0393%2F22%2F5%2F055001>.
- [100] D. D. Symons and N. A. Fleck. The Imperfection Sensitivity of Isotropic Two-Dimensional Elastic Lattices. *Journal of Applied Mechanics*, 75(5): 051011, 2008. doi: 10.1115/1.2913044.

- [101] E. B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Philosophical Magazine A*, 73(6):1529–1563, 1996. doi: 10.1080/01418619608243000. URL <https://doi.org/10.1080/01418619608243000>.
- [102] E. B. Tadmor, G. S. Smith, N. Bernstein, and E. Kaxiras. Mixed finite element and atomistic formulation for complex crystals. *Phys. Rev. B*, 59:235–245, Jan 1999. doi: 10.1103/PhysRevB.59.235. URL <https://link.aps.org/doi/10.1103/PhysRevB.59.235>.
- [103] Thomas Tancogne-Dejean and Dirk Mohr. Elastically-isotropic elementary cubic lattices composed of tailored hollow beams. *Extreme Mechanics Letters*, 22:13 – 18, 2018. ISSN 2352-4316. doi: <https://doi.org/10.1016/j.eml.2018.04.005>. URL <http://www.sciencedirect.com/science/article/pii/S2352431618300245>.
- [104] Thomas Tancogne-Dejean, Adriaan B. Spierings, and Dirk Mohr. Additively-manufactured metallic micro-lattice materials for high specific energy absorption under static and dynamic loading. *Acta Materialia*, 116:14 – 28, 2016. ISSN 1359-6454. doi: <https://doi.org/10.1016/j.actamat.2016.05.054>. URL <http://www.sciencedirect.com/science/article/pii/S1359645416304153>.
- [105] Thomas Tancogne-Dejean, Marianna Diamantopoulou, Maysam Gorji, Colin Bonatti, and Dirk Mohr. 3d plate-lattices: An emerging class of low-density metamaterial exhibiting optimal isotropic stiffness. *Advanced Materials*, 30:1803334, 09 2018. doi: 10.1002/adma.201803334.
- [106] H. C. Tankasala, V. S. Deshpande, and N. A. Fleck. 2013 Koiter Medal Paper: Crack-Tip Fields and Toughness of Two-Dimensional Elastoplastic Lattices. *Journal of Applied Mechanics*, 82(9):091004, September 2015. doi: 10.1115/1.4030666.
- [107] H.C. Tankasala, V.S. Deshpande, and N.A. Fleck. Tensile response of elastoplastic lattices at finite strain. *Journal of the Mechanics and Physics of Solids*, 109:307 – 330, 2017. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2017.02.002>. URL <http://www.sciencedirect.com/science/article/pii/S0022509616307967>.
- [108] I. Tembhekar, J. S. Amelang, L. Munk, and D. M. Kochmann. Automatic adaptivity in the fully nonlocal quasicontinuum method for coarse-grained atomistic simulations. *International Journal for Numerical Methods in Engineering*, 110(9):878–900, 2017. ISSN 1097-0207. doi: 10.1002/nme.5438. URL <http://dx.doi.org/10.1002/nme.5438>. nme.5438.

- [109] Prasanna Thiyagasundaram, Junqiang Wang, Bhavani V. Sankar, and Nagaraj K. Arakere. Fracture toughness of foams with tetrakaidecahedral unit cells using finite element based micromechanics. *Engineering Fracture Mechanics*, 78(6):1277 – 1288, 2011. ISSN 0013-7944. doi: <https://doi.org/10.1016/j.engfracmech.2011.01.003>. URL <http://www.sciencedirect.com/science/article/pii/S0013794411000075>.
- [110] N. Triantafyllidis and S. Bardenhagen. On higher order gradient continuum theories in 1-d nonlinear elasticity. derivation from and comparison to the corresponding discrete models. *Journal of Elasticity*, 33(3):259–293, 1993. ISSN 1573-2681. doi: 10.1007/BF00043251. URL <http://dx.doi.org/10.1007/BF00043251>.
- [111] G. Venturini, J. Z. Yang, M. Ortiz, and J. E. Marsden. Replica time integrators. *International Journal for Numerical Methods in Engineering*, 88(6):586–611, 2011. doi: 10.1002/nme.3202. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.3202>.
- [112] Andrea Vigliotti and Damiano Pasini. Stiffness and strength of tridimensional periodic lattices. *Computer Methods in Applied Mechanics and Engineering*, 229-232:27 – 43, 2012. ISSN 0045-7825. doi: <https://doi.org/10.1016/j.cma.2012.03.018>. URL <http://www.sciencedirect.com/science/article/pii/S0045782512000941>.
- [113] Andrea Vigliotti, Vikram S. Deshpande, and Damiano Pasini. Non linear constitutive models for lattice materials. *Journal of the Mechanics and Physics of Solids*, 64:44 – 60, 2014. ISSN 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2013.10.015>. URL <http://www.sciencedirect.com/science/article/pii/S0022509613002238>.
- [114] Jong Youn Park, YoungâSam Cho, Sung Youb Kim, Sukky Jun, and Seyoung Im. A quasicontinuum method for deformations of carbon nanotubes. *CMES - Computer Modeling in Engineering and Sciences*, 11, 02 2006.
- [115] Xiaoyu Zheng, William Smith, Julie Jackson, Bryan Moran, Huachen Cui, Da Chen, Jianchao Ye, Nicholas Fang, Nicholas Rodriguez, Todd Weisgraber, and Christopher M Spadaccini. Multiscale metallic metamaterials. *Nature Materials*, 15:1100, jul 2016. URL <https://doi.org/10.1038/nmat4694><http://10.0.4.14/nmat4694><https://www.nature.com/articles/nmat4694#supplementary-information>.

A p p e n d i x A

EFFECTIVE ELASTIC CONSTANTS OF PERIODIC TRUSS LATTICES

For comparison, we here report the homogenized stiffness tensors of the various 2D truss lattices in Voigt notation as functions of the relative density $\bar{\rho}$. These stiffness tensors were calculated by solving the unit cell problem with periodic boundary conditions, i.e., we compute

$$\mathbb{C}_{ijkl} = \frac{W}{\partial \varepsilon_{ij} \partial \varepsilon_{kl}}, \quad (\text{A.1})$$

where

$$W(\varepsilon) = \inf_{\mathbf{U}, \boldsymbol{\theta}} \left\{ \frac{1}{|\Omega_{\text{UC}}|} \sum_{e \in \mathcal{E}} W_e(\mathbf{U}, \boldsymbol{\theta}) \mid \mathbf{U}^+ - \mathbf{U}^- = \varepsilon(\mathbf{X}^+ - \mathbf{X}^-), \boldsymbol{\theta}^+ - \boldsymbol{\theta}^- = 0 \right\}, \quad (\text{A.2})$$

where $+$ and $-$ represent nodes on opposite sides of the UC boundary $\partial\Omega_{\text{UC}}$, $(\mathbf{X}^+ - \mathbf{X}^-)$ is the vector that points from the $-$ side to the $+$ side between pairs of boundary nodes, \mathbf{U} and $\boldsymbol{\theta}$ denote the vectors of displacement and rotational dofs, respectively. The thus obtained effective stiffness tensors of the lattice architectures in this study are listed below (with E denoting the base material's Young modulus and $\bar{\rho}$ the relative density). Unfortunately, the analogous expressions for the 3D truss lattices could not be computed in closed form due to their complexity.

A.0.1 Triangle lattice

$$\mathbb{C}_{\text{Voigt}} = \begin{pmatrix} \frac{E\bar{\rho}(\bar{\rho}^2+36)}{96} & -\frac{E\bar{\rho}(\bar{\rho}^2-12)}{96} & 0 \\ -\frac{E\bar{\rho}(\bar{\rho}^2-12)}{96} & \frac{E\bar{\rho}(\bar{\rho}^2+36)}{96} & 0 \\ 0 & 0 & \frac{E\bar{\rho}(\bar{\rho}^2+12)}{96} \end{pmatrix}$$

A.0.2 2D kagome lattice

$$\mathbb{C}_{\text{Voigt}} = \begin{pmatrix} \frac{E\bar{\rho}(\bar{\rho}^2+18)}{48} & -\frac{E\bar{\rho}(\bar{\rho}^2-6)}{48} & 0 \\ -\frac{E\bar{\rho}(\bar{\rho}^2-6)}{48} & \frac{E\bar{\rho}(\bar{\rho}^2+18)}{48} & 0 \\ 0 & 0 & \frac{E\bar{\rho}(\bar{\rho}^2+6)}{48} \end{pmatrix}$$

A.0.3 Square Lattice

$$\mathbb{C}_{\text{Voigt}} = \begin{pmatrix} \frac{E\bar{\rho}}{2} & 0 & 0 \\ 0 & \frac{E\bar{\rho}}{2} & 0 \\ 0 & 0 & \frac{E\bar{\rho}^3}{4} \end{pmatrix}$$

A.0.4 Hexagon lattice

$$\mathbb{C}_{\text{Voigt}} = \begin{pmatrix} \frac{E\bar{\rho}(9\bar{\rho}^2+4)}{4(3\bar{\rho}^2+4)} & \frac{E\bar{\rho}(4-3\bar{\rho}^2)}{4(3\bar{\rho}^2+4)} & 0 \\ \frac{E\bar{\rho}(4-3\bar{\rho}^2)}{4(3\bar{\rho}^2+4)} & \frac{E\bar{\rho}(9\bar{\rho}^2+4)}{4(3\bar{\rho}^2+4)} & 0 \\ 0 & 0 & \frac{3E\bar{\rho}^3}{6\bar{\rho}^2+8} \end{pmatrix}$$

A.0.5 Star lattice

$$\mathbb{C}_{\text{Voigt}} = \begin{pmatrix} \frac{8E\bar{\rho}^3}{16\bar{\rho}^2+25} & 0 & 0 \\ 0 & \frac{8E\bar{\rho}^3}{16\bar{\rho}^2+25} & 0 \\ 0 & 0 & \frac{E\bar{\rho}^3}{\bar{\rho}^2+100} \end{pmatrix}$$

Appendix B

CALCULATING SAMPLE UNIT CELL WEIGHTS IN 3D

In two dimensions, the number of unit cells that lie on element edges can efficiently be calculated when working in Bravais coordinates. The task of counting unit cells on an edge of the element is equivalent to counting the number of lattice points that intersect a line with integer end points $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$, which can be computed with

$$w_{s,\text{edge}}^{(2D)} = \text{gcd}(b_1^{(2)} - b_1^{(1)}, b_2^{(2)} - b_2^{(1)}), \quad (\text{B.1})$$

where $\text{gcd}(z_1, z_2)$ is the greatest common divisor of z_1 and z_2 , $b_k^{(j)}$ represents the k^{th} coordinate of the j^{th} endpoint.

Counting the the number of unit cells on the edges of tetrahedra can similarly be expressed as the number of integer points intersecting a line connecting two points $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ on the 3D integer lattice (corresponding to the unit cell locations of the element vertices in Bravais coordinates). This is given by the expression

$$w_{s,\text{edge}}^{(3D)} = \text{gcd}(b_1^{(2)} - b_1^{(1)}, b_2^{(2)} - b_2^{(1)}, b_3^{(2)} - b_3^{(1)}). \quad (\text{B.2})$$

Counting the number of unit cells on faces of tetrahedra in 3D is more challenging. When the unit cells are expressed in terms of their Bravais coordinates, the faces of the tetrahedra correspond to triangles with integer coordinates embedded in a 3D space. These triangles lie on a planar lattice Λ defined by the intersection of the plane of the triangle and the 3D integer lattice, but $\Lambda \neq \mathbb{Z}^d$. In order to compute the number of lattice points in Λ that lie on the interior of the triangle face, it is first necessary to find basis vectors for Λ , then perform another change of coordinates using the basis vectors of Λ to express the triangle as an integer triangle on an integer lattice. Then computing the number of unit cells that lie on the face can be computed using the same methods that are used to compute the inner sample unit cell weight in the 2D summation rule (i.e. using Pick's theorem).

In order to have cleaner notation, we note that an integer triangle in 3D space (i.e. a face of a tetrahedron element in Bravais coordinates) can trivially be

translated to the origin such that it has vertices $\{\mathbf{0}, \mathbf{Y}^{(2)}, \mathbf{Y}^{(3)}\}$ ($\mathbf{Y}^{(2)}, \mathbf{Y}^{(3)} \in \mathbb{Z}^3$) without changing the number of integer points that intersect the face. The triangle lies on the plane which can be defined by the normal vector

$$\mathbf{n} = \mathbf{Y}^{(2)} \times \mathbf{Y}^{(3)}. \quad (\text{B.3})$$

The lattice on the plane of the triangle face is described by all of the integer solutions of the equation

$$\mathbf{n} \cdot \mathbf{Y} = 0, \quad \mathbf{Y} \in \mathbb{Z}^3, \quad (\text{B.4})$$

which is a linear diophantine equation that can be solved with a known algorithm (see [74]). The solutions to this equation can be parametrized by two non-unique vectors $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \mathbb{Z}^3$ (i.e. basis vectors for the lattice) such that

$$\mathbf{Y} = Z_1 \boldsymbol{\lambda}_1 + Z_2 \boldsymbol{\lambda}_2, \quad Z_1, Z_2 \in \mathbb{Z} \quad (\text{B.5})$$

are all of the solutions to Eq. (B.4). The first basis vector can be chosen to be

$$\boldsymbol{\lambda}_1 = \frac{\{-n_2, n_1, 0\}}{\gcd(n_2, n_1)}. \quad (\text{B.6})$$

The second basis vector can be computed by first finding \tilde{Y}_1 and \tilde{Y}_2 such that

$$n_1 \tilde{Y}_1 + n_2 \tilde{Y}_2 = \gcd(n_1, n_2) = \tilde{n} \quad (\text{B.7})$$

using the Euclidean algorithm, with n_i being the components of \mathbf{n} . Then it can be seen that the second basis vector can be written as

$$\boldsymbol{\lambda}_2 = \frac{\{n_3 \tilde{Y}_1, n_3 \tilde{Y}_2, -\gcd(n_1, n_2)\}}{\gcd(n_3 \tilde{Y}_1, n_3 \tilde{Y}_2, n_1, n_2)}. \quad (\text{B.8})$$

Then, the vertices of the triangle face can be expressed with the integer coordinates on the integer lattice, $\{\mathbf{0}, \mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\}$, with $\mathbf{Z}^{(i)} \in \mathbb{Z}^2$ satisfying

$$\mathbf{Y}^{(i)} = Z_1^{(i)} \boldsymbol{\lambda}_1 + Z_2^{(i)} \boldsymbol{\lambda}_2. \quad (\text{B.9})$$

Finally, the equations (B.1) and (3.62) can be used to compute the number of integer points on the interior of the triangle, which corresponds to the weight of the sampling unit cell on the tetrahedron face.